METHODEN FÜR DEN ENTWURF LOGISCHER DATENBANKSTRUKTUREN

Dissertation
zur Erlangung des Grades
eines Doctor rerum politicarum
in der Abteilung Wirtschafts- und Sozialwissenschaften
der Universität Dortmund

vorgelegt von
Diplom-Kaufmann Astrid Merkle
Ulm 1982

1983

METHODEN FÜR DEN ENTWURF LOGISCHER DATENBANKSTRUKTUREN

Dissertation
zur Erlangung des Grades
eines Doctor rerum politicarum
in der Abteilung Wirtschafts- und Sozialwissenschaften
der Universität Dortmund

vorgelegt von
Diplom-Kaufmann Astrid Merkle
Ulm 1982

Vorwort

Bei Herrn Prof. Dr. J. Griese bedanke ich mich für die Betreuung und Förderung der Arbeit und bei Herrn Prof. Dr. H.-D. Ehrich für seine Anregungen.
Herrn Prof. Dr. Dr. K. Hansen danke ich für die zahlreichen Empfehlungen und Ratschläge und Frau A. Scharpenberg für ihre unermüdliche Nachrichtenübermittlung.

UNIV. GIESSEN BIBL.

1. Gutachter: Prof. Dr. J. Griese
2. Gutachter: Prof. Dr. H.-D. Ehrich

Tag der mündlichen Prüfung: 15.4.1983

INHALTSVERZEICHNIS

		Seite
1.	VORBEMERKUNGEN ZUR STRUKTURIERUNG DER DATEN BEIM DATENBANKENTWURF	1
1.1	EINFÜHRUNG	1
1.1.1	Logische Datenorganisation	1
1.1.2	Objekte und Objekt-Typen	2
1.1.3	Attribute und Attributwerte	2
1.1.4	Datenmodelle	3
1.1.5	Funktionale, voll funktionale, transitive Abhängigkeit und multivalued dependency	5
1.1.6	Normalisierung von Relationen	6
1.2	PROBLEMSTELLUNG UND AUFBAU DER ARBEIT	9

Seite

2. UNTERSUCHUNG IN DER LITERATUR BEHANDELETER METHODEN 12

2.1 ANSAETZE, DIE AUF EINER DATENVERWENDUNGSMATRIX AUFBAUEN 12

2.1.1 Allgemeines 12

2.1.2 Graphentheoretischer Ansatz der Clusteranalyse nach Freudenthaler und Maier 14

2.1.2.1 Zielsetzung 14

2.1.2.2 Vorgehensweise 15

2.1.2.2.1 Bestimmung der Ähnlichkeitsmatrix W 15

2.1.2.2.2 Bestimmung des bestmöglichen Klassifikationsniveaus 17

2.1.2.2.3 Klassifikation der Matrix A durch Subgraphenbildung 18

2.1.2.2.4 Bestimmung disjunkter Metaklassen 20

2.1.2.2.5 Spaltenweise Klassifikation 21

2.1.2.3 Zusammenfassung 23

2.1.3 Partitionierender Ansatz der Clusteranalyse nach Späth 24

2.1.3.1 Zielsetzung 24

2.1.3.2 Vorgehensweise 26

2.1.3.2.1 Bestimmung der Abstandsmatrix 26

2.1.3.2.1.1 Abstandsmatrix mit dem Tanimoto-Abstand 26

2.1.3.2.1.2 Abstandsmatrix mit dem Matching-Koeffizienten 28

2.1.3.2.2 Ermittlung des Homogenitätsmaßes 29

2.1.3.2.3 Optimierung durch das Austausch-Verfahren 30

2.1.3.3 Ergebnis des Austauschverfahrens am Beispiel der Matrix A 31

		Seite
2.1.4	Betrachtung weiter Ähnlichkeitsmaße	33
2.1.4.1	Der S-Koeffizient	33
2.1.4.2	Erste Variante des S-Koeffizienten	35
2.1.4.3	Zweite Variante des S-Koeffizienten	36
2.1.4.4	Dritte Variante des S-Koeffizienten	37
2.1.4.5	Zusammenfassung	38
2.1.5	Clusterbildung unter Berücksichtigung der Zugriffsanforderungen und Normalisierung nach Döringer	39
2.1.5.1	Zielsetzung	39
2.1.5.2	Vorgehensweise	40
2.1.5.2.1	Festlegung der Zugriffs- und Update-Anforderungen	40
2.1.5.2.2	Bildung von Relationen und deren Normalisierung	41
2.1.5.2.3	Zusammenfassung von Relationen in 3NF und Gliederung in Segmente	42
2.1.5.3	Zusammenfassung	43
2.1.6	Clusterbildung durch Berücksichtigung der Reihenfolge der Ordnungskriterien nach Koreimann	44
2.1.6.1	Zielsetzung	44
2.1.6.2	Vorgehensweise	45
2.1.6.2.1	Anwendungsanalyse	45
2.1.6.2.2	Redundanzanalyse	46
2.1.6.2.3	Bildung von Anwendungsfamilien	47
2.1.6.2.4	Strukturierung der Datenbank	49
2.1.6.2.5	Simulation und Test	49
2.1.6.3	Zusammenfassung	50
2.1.7	Gegenüberstellung von Rechenergebnissen	51
2.1.7.1	Allgemeines	51
2.1.7.2	Ergebnisse am Beispiel der Matrix A	52
2.1.7.3	Ergebnisse am Beispiel der Matrix B	54
2.1.7.4	Ergebnisse am Beispiel der Matrix D	55
2.1.7.5	Ergebnisse am Beispiel der Matrix K	57
2.1.7.6	Zusammenfassung	58

Seite

2.2 BESTIMMUNG VON RELATIONEN DURCH GENERALISIERUNG 59

2.2.1 Zielsetzung 59

2.2.2 Vorgehensweise 60
2.2.2.1 Bildung disjunkter Objektklassen 60
2.2.2.2 Bildung von Relationen 62
2.2.2.3 Bestimmung einer allgemeingültigen Struktur 63
2.2.3 Zusammenfassung 64

2.3 RELATIONEN-SYNTHESE 65

2.3.1 Allgemeines 65

2.3.2 Synthetischer Ansatz nach Wang und Wedekind 67
2.3.2.1 Zielsetzung 67
2.3.2.2 Vorgehensweise 67
2.3.2.2.1 Festlegen der Elementaren Funktionalrelationen 68
2.3.2.2.2 Bestimmung der maximalen Distanz einer Relation 70
2.3.2.2.3 Bestimmung der transitiven Abhängigkeiten 71
2.3.2.2.4 Ermittlung der minimalen Überdeckung 72
2.3.2.2.5 Überprüfung der Zugriffshäufigkeiten beim Änderungsdienst 76
2.3.2.3 Zusammenfassung 77

2.3.3 Synthetischer Ansatz nach Bernstein 78
2.3.3.1 Zielsetzung 78
2.3.3.2 Vorgehensweise 79
2.3.3.2.1 Algorithmus zur Ermittlung nichtredundanter Relationen 79
2.3.3.2.2 Behandlung nichtfunktionaler Abhängigkeiten 81
2.3.3.2.3 Ermittlung der optimalen 3NF 82
2.3.3.3 Zusammenfassung 83

		Seite
2.3.4	Synthetischer Ansatz nach Fagin	84
2.3.4.1	Zielsetzung	84
2.3.4.2	Vorgehensweise	85
2.3.4.2.1	Zum Begriff "multivalued-dependencies"	85
2.3.4.2.2	Normalisierung der "multivalued-dependencies"	86
2.3.4.2.3	Beispiel für eine Normalisierung in 4NF	87
2.3.4.3	Zusammenfassung	88
2.3.5	Synthetischer Ansatz nach Vetter	89
2.3.5.1	Zielsetzung	89
2.3.5.2	Vorgehensweise	89
2.3.5.2.1	Festlegung der Ausgangs-Relationen	89
2.3.5.2.2	Behandlung von m:n-Verbindungen	90
2.3.5.2.3	Bestimmung der direkten und indirekten Abhängigkeiten	91
2.3.5.2.4	Eliminieren von transitiven Abhängigkeiten	92
2.3.5.2.5	Bestimmung der optimalen 3NF	94
2.3.5.2.6	Überprüfung der Zugriffsanforderung	95
2.3.5.3	Zusammenfassung	95
2.3.6	Gegenüberstellung der Rechenergebnisse	96
2.3.6.1	Rechenergebnisse des ersten Beispiels	96
2.3.6.1.1	Ermittlung der 3NF nach dem Verfahren von Wang und Wedekind	96
2.3.6.1.2	Ermittlung der 3NF nach dem Verfahren von Vetter	99
2.3.6.2	Rechenergebnisse des zweiten Beispiels	101
2.3.6.2.1	Ermittlung der 3NF nach dem Verfahren von Wang und Wedekind	101
2.3.6.2.2	Ermittlung der 3NF nach dem Verfahren von Vetter	109
2.3.6.3	Rechenergebnisse des dritten Beispiels	111
2.3.6.3.1	Ermittlung der 3NF nach dem Verfahren von Wang und Wedekind	111
2.3.6.3.2	Ermittlung der 3NF nach dem Verfahren von Vetter	112
2.3.6.4	Zusammenfassung	113

Seite

3. STRUKTURIERUNG DES VORHANDENEN DATENMATERIALS FÜR EIN RELATIONALES DATENBANKSYSTEM 114

3.1 ZIELSETZUNG 114

3.2 SYSTEMANALYSE 116
3.2.1 Allgemeine Vorgehensweise beim Datenbank-Design 116
3.2.2 Orientierung über das Problem 117
3.2.3 Wahl passender Entscheidungskriterien 120
3.2.4 Formulierung der Entscheidungsalternativen 121
3.2.5 Auswahl relevanter Umweltfaktoren 122
3.2.5.1 Analyse der entities und ihrer Ausprägungen 122
3.2.5.2 Analyse der funktionalen Abhängigkeiten 124
3.2.5.2.1 Bestimmung der m:n Verbindungen 124
3.2.5.2.2 Bestimmung der Elementaren Funktionalrelationen 127

3.3 MODELLKONSTRUKTION 129
3.3.1 Erzeugen von Relationen in 3NF 129
3.3.2 Minimierung der Anzahl der Relationen 132
3.3.3 Ausschließen von Zyklen 133
3.3.4 Berücksichtigung der Häufigkeit des Datenzugriffs 134
3.3.4.1 Mögliche Alternativen 134
3.3.4.2 Bewertung der Kanten des Graphen 134
3.3.4.3 Bewertung der Knoten des Graphen 135
3.3.4.3.1 Einfluß der Knotennumerierung auf den Baumalgorithmus 135
3.3.4.3.2 Einfluß der Knotennumerierung auf die Zyklenauflösung 136
3.3.4.3.3 Unerwünschte Beeinträchtigung der Lösung durch Entfernen einer bestimmten Kante im Zyklus 137
3.3.4.4 Zusammenfassung 138
3.3.5 Maschinelle Überprüfung der Ergebnisse 139

		Seite
3.4	MODELLVALIDIERUNG ANHAND VON BEISPIELEN	140
3.4.1	Überprüfung des ersten Beispiels	140
3.4.1.1	Charakteristische Merkmale des 1.Beispiels	140
3.4.1.2	Analyse der Ergebnisse des 1. Beispiels	141
3.4.1.2.1	Überprüfung der Ergebnisse auf optimale 3NF	141
3.4.1.2.2	Überprüfung der Erreichbarkeit der Knoten	141
3.4.1.2.2.1	Erreichbarkeit im Vergleich mit dem zyklenfreien Graphen	141
3.4.1.2.2.2	Erreichbarkeit im Vergleich zum Graphen mit Zyklen	141
3.4.1.2.3	Berücksichtigung der Häufigkeit des Datenzugriffs und deren Auswirkung	142
3.4.2	Überprüfung des zweiten Beispiels	143
3.4.2.1	Charakteristische Merkmale des 2.Beispiels	143
3.4.2.2	Analyse der Ergebnisse des 2. Beispiels	144
3.4.2.2.1	Überprüfung der Ergebnisse auf optimale 3NF	144
3.4.2.2.2	Überprüfung der Erreichbarkeit der Knoten	144
3.4.2.2.2.1	Erreichbarkeit im Vergleich mit dem zyklenfreien Graphen	144
3.4.2.2.2.2	Erreichbarkeit im Vergleich zum Graphen mit Zyklen	145
3.4.2.2.3	Berücksichtigung der Häufigkeit des Datenzugriffs und deren Auswirkung	146
3.4.3	Überprüfung des dritten Beispiels	147
3.4.3.1	Charakteristische Merkmale des 3.Beispiels	147
3.4.3.2	Analyse der Ergebnisse des 3. Beispiels	147
3.4.3.2.1	Überprüfung der Ergebnisse auf optimale 3NF	147
3.4.3.2.2	Überprüfung der Erreichbarkeit der Knoten	148
3.4.3.2.2.1	Erreichbarkeit im Vergleich mit dem zyklenfreien Graphen	148
3.4.3.2.2.2	Erreichbarkeit im Vergleich zum Graphen mit Zyklen	148
3.4.3.2.3	Berücksichtigung der Häufigkeit des Datenzugriffs und deren Auswirkung	148

Seite

3.4.4 Überprüfung des vierten Beispiels 149
3.4.4.1 Charakteristische Merkmale des 4.Beispiels 149
3.4.4.2 Analyse der Ergebnisse des 4. Beispiels 150
3.4.4.2.1 Überprüfung der Ergebnisse auf optimale 3NF 150
3.4.2.2.2 Überprüfung der Erreichbarkeit der Knoten 150
3.4.2.2.2.1 Erreichbarkeit im Vergleich mit dem zyklenfreien Graphen 150
3.4.2.2.2.2 Erreichbarkeit im Vergleich zum Graphen mit Zyklen 150
3.4.2.2.3 Berücksichtigung der Häufigkeit des Datenzugriffs und deren Auswirkung 150
3.4.5 Überprüfung des fünften Beispiels 151
3.4.5.1 Charakteristische Merkmale des 5.Beispiels 151
3.4.5.2 Analyse der Ergebnisse des 5. Beispiels 152
3.4.5.2.1 Überprüfung der Ergebnisse auf optimale 3NF 152
3.4.5.2.2 Überprüfung der Erreichbarkeit der Knoten 152
3.4.5.2.2.1 Erreichbarkeit im Vergleich mit dem zyklenfreien Graphen 152
3.4.5.2.2.2 Erreichbarkeit im Vergleich zum Graphen mit Zyklen 152
3.4.5.2.3 Berücksichtigung der Häufigkeit des Datenzugriffs und deren Auswirkung 152
3.5 DEDUKTION VON INFORMATIONEN AUS DEM MODELL 153

3.6 ALLGEMEINGÜLTIGE MODELLVALIDIERUNG 155
3.6.1 Überprüfung der optimalen 3NF 155
3.6.2 Überprüfung der maximalen Erreichbarkeit der Knoten untereinander 157
3.6.2.1 Bei gerichteten Graphen ohne Zyklen 157
3.6.2.2 Bei gerichteten Graphen mit Zyklen 159
3.6.2.2.1 LP-Ansatz für das erste Beispiel 159
3.6.2.2.2 LP-Ansatz für das zweite Beispiel 160
3.6.2.2.3 LP-Ansatz für das dritte Beispiel 161
3.6.2.2.4 LP-Ansatz für das vierte Beispiel 162
3.6.2.2.5 LP-Ansatz für das fünfte Beispiel 163
3.6.3 Verfahrensvergleich 164

4. SCHLUßBETRACHTUNG 168

VERZEICHNIS DER ABBILDUNGEN

Bild 1	Matrix A
Bild 2	Ähnlichkeitsmatrix W
Bild 3	Ähnlichkeitsmatrix unter Berücksichtigung des Klassifikationsniveaus $w_x = 2$
Bild 4	Graph für $w_x = 2$
Bild 5	Matrix A zeilenklassifiziert
Bild 6	transponierte Matrix A
Bild 7	Ähnlichkeitsmatrix W
Bild 8	Ähnlichkeitsmatrix unter Berücksichtigung des Klassifikationsniveaus $w_x = 2$
Bild 9	Matrix A zeilen- und spaltenklassifiziert
Bild 10	Abstandsmatrix mit dem Matching-Koeffizienten
Bild 11	Matrix A
Bild 12	Ähnlichkeitsmatrix mit dem S-Koeffizienten
Bild 13	Ähnlichkeitsmatrix für die transponierte Matrix A mit dem S-Koeffizienten
Bild 14	Datenverwendungsmatrix nach Döringer
Bild 15	Gruppierung der Anforderungen
Bild 16	Beispiel eines Datenkatalogs
Bild 17	Beispiel für Daten einer Anwendungsfamilie
Bild 18	Hierarchische Klassifikation
Bild 19	Klassifikation ohne disjunkte Klassen
Bild 20	Matrix M
Bild 21	Matrix R
Bild 22	Matrix T
Bild 23	Matrix W
Bild 24	Matrix äquivalenter Relationen
Bild 25	reduzierte Matrix äquivalenter Relationen
Bild 26	Matrix R für das erste Beispiel
Bild 27	Matrix T für das erste Beispiel
Bild 28	Adjazenzmatrix für das zweite Beispiel
Bild 29	maximale Distanzen für das zweite Beispiel
Bild 30	Matrix R für das zweite Beispiel

ABKÜRZUNGSVERZEICHNIS

1NF	erste Normalform
2NF	zweite Normalform
3NF	dritte Normalform
4NF	vierte Normalform
BCNF	Boyce Codd Normalform
EFR	Elementare Funktionalrelation
FA	funktional abhängig
MVD	multivalued dependency
NSA	Nichtschlüsselattribut
SA	Schlüsselattribut
SZK	Subgraphenzentrumsknoten
TA	transitiv abhängig
VFA	voll funktional abhängig

SYMBOLVERZEICHNIS

Allgemein:

$\Rightarrow$	voll funktional abhängig
$\rightarrow$	funktional abhängig
$\not\Rightarrow$	nicht voll funktional abhängig
$\not\rightarrow$	nicht funktional abhängig
$\rightarrow\!\!\!\rightarrow$	multivalued dependencies

zu Kapitel 2.1:

a_{ij}	Elemente der Datenverwendungsmatrix
b_k	Anzahl der Elemente des k-ten Clusters
b_{ij}	Anzahl der Elemente für die gilt $a_{iq}=0$, $a_{jq}=1$ (1)
c_{ij}	Anzahl der Elemente für die gilt $a_{iq}=1$, $a_{jq}=0$ (1)
d_{ij}	Anzahl der übereinstimmenden 1-Merkmale (1)
	(1) bzw. Summe der Gewichte G_i...
$C_1..C_l$	Cluster 1 $\leq k \leq$ l mit l=Anzahl der Cluster
D	Zielfunktion des Austauschverfahrens nach Späth
e_k	Einzelhomogenität des k-ten Cluster
i	Zeilen der Matrix
j	Spalten der Matrix
l	Anzahl der Cluster
m	Anzahl der Zeilen der Datenverwendungsmatrix
n	Anzahl der Spalten der Datenverwendungsmatrix
t_{ij}	Abstandsmaß
T	Abstandsmatrix
w_{ij}	Ähnlichkeitsmaß
W	Ähnlichkeitsmatrix
z_i	Zeilensumme der Matrix

SYMBOLVERZEICHNIS

zu Kapitel 3:

E	eine genügend große Distanz von der Wurzel zum Endknoten des Graphen
(i,j)	Pfeil eines Digraphen mit dem Anfangsknoten i und dem Endknoten j
m	Anzahl der Kanten des Graphen X = (K,E)
M	Eine genügend große Zahl
n	Anzahl der Knoten des Graphen X = (K,E)
N(j)	Menge der Nachfolgerknoten eines Knoten j
D = (K,E)	Digraph oder gerichteter Graph mit der Knotenmenge K und der Pfeilmenge E
u_i	Maximale Weglänge von der Wurzel eines Graphen zum Knoten i
V(j)	Menge der Vorgängerknoten eines Knoten j
x_{ij}	Kante eines Graphen von Knoten i nach j
X = (K,E)	Ungerichteter Graph mit der Knotenmenge K und der Pfeilmenge E

1. VORBEMERKUNGEN ZUR STRUKTURIERUNG DER DATEN BEIM DATENBANKENTWURF

1.1 EINFÜHRUNG

In der vorliegenden Arbeit werden Methoden zur Unterstützung des logischen Datenbankdesign untersucht und es wird eine neue Methode zur Strukturierung des Datenmaterials entwickelt. Dabei wird vorausgesetzt, daß der Leser vertraut ist mit den Begriffen, die im Zusammenhang mit der Konzeption von Datenbanken - insbesondere der relationalen Datenmodelle - stehen. Daher werden nachstehend nur die wichtigsten Begriffe definiert um einen einheitlichen Sprachgebrauch sicherzustellen.

1.1.1 Logische Datenorganisation

Der Systemaufbau von Datenbankmodellen läßt sich in folgende Ebenen gliedern:[1)]

- Ebene der Datenstrukturen
  (logische Sicht des Benutzers,
  user's logical view)

- Ebene der Zugriffspfade
  (logische Sicht des Systems,
  system logical view)

- Ebene der physischen Speicherung
  auf den Geräten

Der amerikanische Normenausschuß ANSI/SPARC sieht folgende drei Schichten vor:[2)]

- das externe Schema
- das konzeptionelle Schema
- das interne Schema

---

1) vgl. Wedekind, H.: Datenbanksysteme I.
Mannheim 1974, S.11

2) vgl. ANSI/x3/SPARC Study Group on Data Base
Management Systems, Interim Report 75-02-08.
In: FDT-Bulletin of ACM-SIGMOD, 7 (1975)
und Härder, Th.: Implementierung von Datenbanksystemen.
München 1978, S.29

Dabei beschreibt das externe Schema die Daten so, wie sie der Benutzer sieht. Das konzeptionelle Schema liefert eine vollständige, redundanzfreie Darstellung der logischen Datenstrukturen aus der Miniwelt des Benutzers, das interne Schema beschreibt die physische Organisation der Daten.
Die logische Datenorganisation umfaßt die externe und die konzeptionelle Ebene.

1.1.2 Objekte und Objekt-Typen

Soll eine Datenbank aufgebaut werden, so müssen die konkreten Phänomene der realen Welt (die gespeichert und wiedergegeben werden sollen) so abstrahiert werden, daß ein geeignetes Modell gebildet werden kann.
Beim Datenbankentwurf wird untersucht welche Beziehungen zwischen den abzubildenden Objekten (entities) der realen Welt bestehen.
Zur Modellbildung werden nach geeigneten Ähnlichkeitskriterien Objekte zu Objekt-Typen (entity-Typen) zusammengefaßt.[3)]

1.1.3 Attribute und Attributwerte

Für die Objekte (entities) der realen Welt müssen die charakteristischen Merkmale (Attribute) bestimmt und deren mögliche Werte (Attributwerte) festgelegt werden.
Wird für die Mitarbeiter eines Unternehmens zum Beispiel der Objekt-Typ "Angestellter" gebildet, so könnten die Merkmale u.a.

- Personalnummer
- Name
- Anschrift
- Berufsbezeichnung

sein.

---

3) vgl. Schlageter, G., Stucky, W.:
Datenbanksysteme: Konzepte und Modelle.
Stuttgart 1977, S.1o ff

Die Werte dieser Merkmale (Attributwerte) könnten dann z.B.

- 007
- Bond, James
- Irgendwo
- Kammerjäger

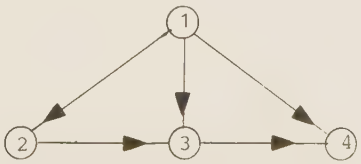
lauten.

1.1.4 Datenmodelle

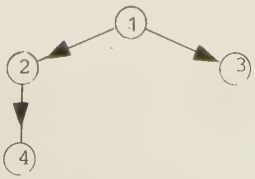
Bei den Datenmodellen unterscheidet man aus Sicht der Benutzer:

- Netzwerk-Datenmodelle
- hierarchische Datenmodelle
- relationale Datenmodelle

Beschreibt man ein abzubildendes Unternehmensmodell durch die Beziehung der entity-Typen, so erhält man einen gerichteten Graphen dessen Knoten die entity-Typen und dessen Kanten die abstrakten Beziehungen zwischen den entity-Typen sind. Beim Netzwerk-Datenmodell sind alle möglichen Beziehungen zwischen den entity-Typen zugelassen, so daß man ein Netzwerk erhält.[4)]



Beim hierarchischen Datenmodell besteht der gerichtete Graph aus disjunkten Teilgraphen. Diese Teilgraphen sind Bäume und der Zugriff erfolgt jeweils über die Wurzel des Baumes.[4)]



4) vgl. Schlageter, G., Stucky, W.: Datenbanksysteme, a.a.O., S. 109 und 117

Das relationale Datenmodell (Relationenmodell) baut auf der Relationen-Theorie auf. Aus der Sicht des Benutzers setzt sich die Datenbank aus Tabellen zusammen. Das relationale Datenmodell bietet einen hohen Grad an Datenunabhängigkeit und scheint sich gegenüber den anderen Datenmodellen auch deshalb durchzusetzen, weil die Notwendigkeit zur Umstrukturierung von Relationen bei Einführung neuer Typen von Daten verringert wird. [5] Aus diesem Grund wird die vorliegende Arbeit auf die Strukturierung der Daten für relationale Datenmodelle eingeschränkt.

Geht man von Attributen A, A_1, A_2, ..., A_n mit den Wertebereichen W, W_1, W_2, ..., W_n aus, so ist eine n-stellige Relation eine Teilmenge des kartesischen Produkts $R \subseteq W_1 \times W_2 \times .... W_n$ über eine gegebene Menge von Bereichen (domains) W_1, W_2, ...W_n.

Dabei ist n der Grad (degree) der Relation.

Ein Element $r = (a_1,a_2,...,a_n) \in R$ ($a_i \in W_i$, i=1,2,..,n) ist ein n-Tupel, also eine Zeile der Tabelle. [6]

Ein entity-Typ ist eindeutig beschrieben durch eine Attributkombination und ein entity von diesem Typ durch eine entsprechende Attributwertkombination.

Einem entity-Typ entspricht ein Relationstyp

R-Typ ($Attribut_1$, $Attribut_2$,...$Attribut_n$)

Dabei ist $Attribut_i$ der Name für das i-te Attribut A_i mit dem Wertebereich W_i.

Nachfolgend wird zwischen Relation und Relationstyp nicht mehr unterschieden.

Das relationale Datenmodell wird bei Schlageter und Stucky folgendermaßen definiert: [6]

> Es gibt eine Menge $\mathcal{R}$ von benannten Relationstypen unterschiedlichen Grades über Attribute $A \in \mathcal{A}$.
> Jede Relation R (eines Typs aus $\mathcal{R}$) besitzt einen Primärschlüssel: $R \hat{=}$ R-Typ (R-Key,$R\text{-}A_1$,...,$R\text{-}A_n$).
> Die Relationen R sind zeitlich nicht konstant.

---

5) vgl. Codd, E.F.: A Relational Model of Data for Large Shared Data Bases. Comm. ACM 13(1970), S.377 ff

6) vgl. Schlageter, G., Stucky, W.: Datenbanksysteme, a.a.O., S. 134 ff
und Wedekind, H.: Datenbanksysteme I, a.a.O., S. 41ff

1.1.5 Funktionale, voll funktionale, transitive Abhängigkeit und multivalued dependency[7)]

Das Attribut B ist funktional abhängig (FA) von dem Attribut A, wenn zu jedem Zeitpunkt jeder Wert von A nicht mehr als einen Wert in B hat.

R (A,B..) sei ein Relationstyp; $A,B \subseteq \mathcal{A}(R)$.
Dann heißt B funktional abhängig von A (in R), in Zeichen: R.A ↠ R.B oder, wenn R eindeutig ist A ↠ B, wenn gilt:

(FA) $\forall R_t$: $\forall x \in W(A)$ $\exists$ höchstens ein $y \in W(B)$: $(x,y..) \in R_t$.

Haben A und B zusätzlich keine gemeinsamen Attribute, so heißt A auch Determinante für B.

Beispiel: Prüfungsnr., Matrikelnr. ↠ Name des Studenten
aber auch Matrikelnr. ↠ Name des Studenten

Ist B funktional abhängig von A, aber nicht funktional abhängig von einer echten Teilmenge von A, so ist B voll funktional abhängig (VFA) von A.

R (A,B..) sei ein Relationstyp; $A,B \subseteq \mathcal{A}(R)$.
Dann heißt B voll funktional abhängig von A (in R), in Zeichen: R.A ⇻ R.B oder A ⇻ B, wenn gilt:

(VFA) $\forall A' \subset A$, $A' \neq A$: R.A' ↛ R.B.

Die Bedingung VFA besagt, daß A eine Determinante für B ist, sofern A und B keine Attribute gemeinsam haben.

Beispiel: wegen Matrikelnr. ↠ Name des Studenten
ist Prüfungsnr.,Matrikelnr. ↛ Name des Studenten
aber Prüfungsnr.,Matrikelnr. ⇻ Note
und Matrikelnr. ⇻ Name des Studenten

R (A,B..) sei ein Relationstyp; $A,B \subseteq \mathcal{A}(R)$.
Dann heißt B transitiv abhängig (TA) von A, wenn gilt:

(TA) $\exists C \subseteq \mathcal{A}(R)$, $B \notin C$: A ↠ C ↠ B, C ↛ A. Daraus folgt:
A ↠ B, aber B ↛ A. A ⇄ C ↠ B

Beispiel:

Die Begriffe transitive Abhängigkeit und transitive Zusammensetzung werden nachfolgend synonym verwendet.

---

7) vgl. Schlageter, G., Stucky, W.: Datenbanksysteme, a.a.O., S. 145 ff
und Wedekind, H.: Datenbanksysteme I, a.a.O., S.55, 58 und 63f

Sei R (A,B,C) eine Relation; A,B,C Attributkombinationen.
Dann besteht zwischen A und B eine multivalued dependency (MVD) in R, in Zeichen: R.A $\twoheadrightarrow$ R.B oder A $\twoheadrightarrow$ B, wenn gilt:

(MVD) $\forall R_t$: $\forall a \in W(A)$, c, c' $\in W(C)$:

$$\{b \in B \mid (a,b,c) \in R_t\} = \{b \in B \mid (a,b,c') \in R_t\} .$$

In Worten: die b-Werte, die a in der Relation R zugeordnet sind, sind unabhängig von Werten von C (die B-Werte wiederholen sich für alle C-Werte).

Beispiel: Arbeitnehmer, Name des(der) Kindes(r), Einkommen

1.1.6 Normalisierung von Relationen[8)]

Die unnormalisierte Form der Relationen enthält im Gegesatz zur ersten Normalform (1NF) Attribute, die selbst noch Relationen sind.
In der zweiten Normalform (2NF) sind nicht voll funktionale Abhängigkeiten zwischen dem Schlüssel- und Nichtschlüsselbereich eliminiert.
Die dritte Normalform (3NF) enthält gegenüber der 2NF keine transitiven Abhängigkeiten mehr.
Eine Relation ist in Boyce-Codd-Normalform (BCNF), wenn sie in 3NF ist und jedes SA und NSA voll funktional abhängig ist von jedem Schlüssel in dem es nicht enthalten ist.
Eine Relation R (A,B,C) mit A $\twoheadrightarrow$ B und A $\Rightarrow$ C kann ohne Informationsverlust in die beiden Relationen S (A,B) und T (A,C) zerlegt werden, dann sind S und T in vierter Normalform (4NF).

Definitionen:
Eine Relation ist in der 2NF, wenn sie in der 1NF ist und wenn jedes Nicht-Primärattribut von R voll funktional abhängig ist von jedem Schlüsselattribut von R.

$R (A_1,\ldots,A_n)$ sei eine Relation, A_i Attribute; $A \subseteq \mathfrak{A}(R)$.

A heißt Schlüsselattribut (SA) oder Primärattribut, wenn gi

(SA) $\exists B \subseteq \mathfrak{A}(R)$, B Schlüssel: $A \in B$.

A heißt Nichtschlüsselattribut (NSA), wenn gilt:

(NSA) kein Attribut $A_i \in A$ ist Schlüsselattribut.

---

8) vgl. Wedekind, H., Ortner, E.: Systematisches Konstruieren von Datenbankanwendungen. München 1980, S. 39
und Schlageter, G., Stucky, W.: Datenbanksysteme, a.a.O., S. 148 ff

Eine Relation ist in zweiter Normalform (2NF), wenn sie in 1NF ist und wenn zusätzlich gilt:

(2NF) $\forall$ A,B $\subseteq \mathcal{A}(R)$: A Schlüssel, B NSA, daraus folgt A $\rightarrow$ B (d.h. jedes NSA ist voll funktional abhängig von jedem Schlüssel).

Die Bedingung der 2NF ist immer dann erfüllt, wenn jeder Schlüssel aus einem einzigen Attribut besteht.

Die Relation R ist in dritter Normalform (3NF), wenn sie in 2NF ist und wenn zusätzlich gilt:

(3NF) $\forall$ A,B $\subseteq \mathcal{A}(R)$: A Schlüssel, B NSA; daraus folgt, daß B nicht transitiv abhängig von A ist.

Eine Relation R ist in Boyce-Codd-Normalform (BCNF), wenn sie in 3NF ist und wenn zusätzlich gilt:

(BCNF) $\forall$ A $\subseteq \mathcal{A}(R)$, B $\varepsilon \mathcal{A}(R) \setminus$A: A $\rightarrow$B; daraus folgt A enthält einen Schlüssel, d.h. jede Determinante enthält einen Schlüssel.

Der Unterschied zur 3NF besteht darin, daß nun auch jedes Schlüsselattribut - nicht nur jedes NSA - voll funktional abhängig von jedem Schlüssel ist, in dem es nicht enthalten ist.

Beispiel: [9] Gegeben seien folgede funktionalen Abhängigkeiten:

Abteilung, Zimmernr. $\rightarrow$ Abteilungsleiter
Abteilungsleiter $\rightarrow$ Zimmernummer
Abteilungsleiter $\rightarrow$ Abteilung

weil Abteilungsleiter Schlüssel ist, ist folgende Relation zwar in 3NF, aber nicht in BCNF:

R(Abteilung, Zimmernr., Abteilungsleiter)

Eine Relation R ist in vierter Normalform 4NF, [10] wenn sie in 3NF ist und wenn zusätzlich gilt:

(4NF) $\forall$ A $\subseteq \mathcal{A}(R)$, B $\varepsilon \mathcal{A}(R) \setminus$A: A $\twoheadrightarrow$B, dann enthält A einen Schlüssel. B ist eine nichtlere Menge.

---

9) vgl. Bernstein, P.A.: Normalization and Functional Dependencies in the Relational Data Base Model. Tech. Rep. CSRG-60, Ph.D.Diss. Computer Systems Research Group. Dep. Computer Science, University of Toronto. Toronto Okt. 1975, S.18

10) vgl. Fagin, R.: Multivalued Dependencies and a New Normal Form for Relational Databases. IBM RJ1812 (26109) IBM Research Laboratory San Jose, Calif. 1976, S.10 ff
und Fagin, R.: Multivalued Dependencies and a New Normal Form for Relational Databases. ACM Trans. on Database Systems. 2 (1977), S. 262 ff

Beispiel: gegeben seien die mulitvalued dependencies:

Arbeitnehmer ↠ Kinder

Arbeitnehmer ↠ Einkommen,Jahr,

dann ist die Relation

Arbeitnehmer, Kinder, Einkommen, Jahr

in BCNF, aber nicht in 4NF, während die Relationen

Arbeitnehmer, Kinder

Arbeitnehmer, Einkommen, Jahr in 4NF sind.

R (A,B...) sei ein Relationstyp; $A,B \subseteq \mathcal{Q}(R)$.

Dann ist R eine Elementare Funktionalrelation (EFR), [11]

wenn B voll funktional abhängig von A ist. $A \rightarrow B$,

Eine funktionale Abhängigkeit $g \in G$ ist redundant, [12]

wenn $g \in (G - \{g\})^+$

Beispiel: gegeben seien die funktionalen Abhängigkeiten:

$A \rightarrow B \rightarrow C$ und $A \rightarrow C$; $A \neq B \neq C$, $B \not\rightarrow A$,

dann ist die funktionale Abhängigkeit $A \rightarrow C$ redundant, da sie transitiv ist und implizit in $A \rightarrow B \rightarrow C$ enthalten ist.

G sei eine Menge Elementarer Funktionalrelationen (EFR),

$$G = \{l_1, l_2, .. l_i, .. l_n\} \; ; l_i = \text{EFR}, \; l_i \neq l_j, \quad \forall \; i,j$$

dann ist $G^+ \subseteq G$ eine minimale Überdeckung, wenn alle redundanten Abhängigkeiten eliminiert wurden. [11] Aus den funktionalen Abhängigkeiten können - durch die Attribute, die in diesen funktionalen Abhängigkeiten enthalten sind - Relationen gebildet werden. Diese Relationen sind in 3NF. Die Anzahl der Relationen einer minimalen Überdeckung ist insoweit minimal, als keine weiteren redundanten Abhängigkeiten eliminiert werden können. Daher wird in der Literatur häufig auch der Begriff der optimalen 3NF anstelle der minimalen Überdeckung verwendet. Dabei handelt es sich um ein lokales und kein globales Optimum. [11] In der vorliegenden Arbeit werden die Begriffe optimale 3NF und minimale Überdeckung synonym verwendet.

---

11) vgl. Delobel, C.: A Theorey about Data in an Information System.IBM Research Report RJ964, San José, 1972,S.24
und Delobel,C., Casey, R.G.: Decomposition of a Data Base and the Theorey of Boolean Switching Functions. IBM Journal of Research and Development. 17 (1973),S.376
und Wedekind, H.: Datenbanksysteme I, a.a.O., S.67 ff

12) vgl. Beeri,D., Bernstein,P.A.,Goodman,N.: A Sophisticate's Introduction to Database Normalization Theorey. In: P of the 3rd Int.Conf. on Very Large Data Bases. New York 1977, S. 115

Der Normalisierungprozeß spielt im Zusammenhang mit relationalen Datenbanken eine große Rolle. Relationen in 3NF, BCNF oder 4NF bieten folgende Vorteile:[13)]

- Vermeidung von Anormalitäten im Änderungsdienst bei den Operationen: einfügen, ändern, löschen
- Vermeidung von Umstrukturierungen beim Einführen neuer Datentypen
- erhöhte Aussagekraft für den Benutzer
- weitgehende Unabhängigkeit von Anfragehäufigkeit und -struktur

Unter Anormalität im Änderungsdienst wird folgendes verstanden: in der Relation

Abteilung (ABT-NR, Abt.-Leiter, Gebäudenr., Hausmeister)

entsteht eine Anormalität beim Einfügen (insertion anormality), wenn ein neues Gebäude gebaut wird, für das zwar der Hausmeister bestimmt wurde, nicht aber die Abteilung, die dort einziehen soll. Eine Anormalität beim Ändern (update anormality) entsteht dadurch, daß die Änderung des Hausmeisters eines Gebäudes in allen Abteilungen dieses Gebäudes berücksichtigt werden muß. Eine Änderung beim Löschen (deletion anormality) entsteht, wenn die letzte Abteilung eines Gebäudes auszieht und dadurch Gebäudenummer und Hausmeister nicht mehr gespeichert sind.

1.2 PROBLEMSTELLUNG UND AUFBAU DER ARBEIT

Gesucht wird eine Methode zur Strukturierung von Daten für ein relationales Datenmodell. Für ein relationales Datenmodell deshalb, weil dieses Datenmodell bei Einführung neuer Attribut-Typen weitgehend frei von Umstrukturierung ist und dadurch immer mehr Verbreitung findet. Da die Daten für ein relationales Datenmodell bestimmt sind, müssen sie mindestens den Anforderungen der optimalen 3NF genügen. Obwohl das Relationenmodell weitgehend unabhängig von der Häufigkeit des Datenzugriffs ist, soll untersucht werden, ob die Zugriffshäufigkeit berücksichtigt werden kann und wenn ja, inwieweit dies sinnvoll ist.

13 vgl. Codd, E.F.: Further Normalization of the Data Base Relational Model. In: Data Base Systems, hrsg. von R. Rustin, Courant Computer Science Symposium 6, May 1971. Prentice-Hall Inc., Englewood Cliffs, N.J. 1972, S. 34
und Schlageter, G., Stucky, W.: Datenbanksysteme, a.a.O., S. 145 und 149

Die Problemstellung wird eingeschränkt auf Relationen deren Schlüssel aus einzelnen Attributen und nicht aus Attributgruppen bestehen und darauf, daß der Graph aus voll funktionalen Abhängigkeiten nicht mehr als eine Wurzel hat.
Bei der Strukturierung der Daten für ein Relationenmodell gibt es drei Hauptprobleme:

- das Auffinden geeigneter Relationen
- die Normalisierung dieser Relationen in 3NF
- das Eliminieren überflüssiger Relationen, so daß eine optimale 3NF erreicht wird.

Um geeignete Relationen aufzufinden, könnte man eine herkömmliche Datenverwendungsmatrix aufstellen, die in den Zeilen alle Attribute und in den Spalten die möglichen Verwendungen enthält.
Die Attribute einer solchen Datenverwendungsmatrix könnten durch Clusteranalyse zu Gruppen zusammengefaßt werden. Für jedes Cluster könnte - aus den darin vorkommenden Attributen - eine unnormalisierte Relation gebildet werden. Die so gefundenen Relationen könnten dann als Eingabe für einen Normalisierungsprozeß bzw. als Eingabe für das neu zu entwickelnde Verfahren dienen.
Aus diesem Grund werden im Kapitel 2.1 einige dieser Verfahren untersucht, die auf einer Datenverwendungsmatrix mit anschließender Clusterung aufbauen. Eines dieser Verfahren - es wird im Kapitel 2.1.5 beschrieben - normalisiert die Relationen, die durch die Clusteranalyse gefunden wurden.
Im Kapitel 2.2 wird ein Verfahren untersucht, das durch Generalisierung und Typisierung geeignete Relationen bildet. Dieses Verfahren baut weder auf einer Datenverwendungsmatrix auf, noch ermittelt es die Relationen durch Untersuchung der funktionalen Abhängigkeiten, vielmehr werden zunächst die charakteristischen Eigenschaften der Attribute untersucht und durch Generalisierung geeignete Attribut-Typen gefunden. Aus diesem Grund wurde hierfür ein eigenes Kapitel vorgesehen.

Eine dritte Gruppe von Verfahren, die im Kapitel 2.3 untersucht werden, beschäftigt sich mit der Relationensynthese. Durch Synthese voll funktionaler Abhängigkeiten werden Relationen gefunden, die den Anforderungen der 3NF, BCNF oder 4NF und denen der minimalen Überdeckung genügen.

Im dritten Kapitel wird eine neue - rechnerunterstützte - Methode entwickelt. Dabei wird die Frage aufgeworfen, inwieweit die minimale Überdeckung ein zweckmäßiges Ergebnis für die Datenstruktur liefert. Insbesondere bei inversen Abhängigkeiten wurden in der Literatur unterschiedliche Lösungen festgestellt auch verursachen Zyklen Probleme bei der Bestimmung der minimalen Überdeckung. Aus diesem Grund wurde folgende Lösung angestrebt: Aus einer gegebenen Menge G von Elementaren Funktionalrelationen l_i

$$G = \left\{l_1, l_2, .. l_i, .. l_n\right\}$$

wird G^+ ermittelt.Dabei soll G^+ in optimaler 3NF sein. Zusätzlich gilt: G^+ enthält keine inversen Abhängigkeiten und es gibt in G^+ keine funktionalen Abhängigkeiten, die einen Zyklus bilden.
Nach Möglichkeit sollen die funktionalen Abhängigkeiten soweit reduziert werden, daß ein Attribut nur noch durch eine funktionale Abhängigkeit bestimmt wird. Dennoch darf die Erreichbarkeit der Attribute nicht verloren gehen. Die Ermittlung von G^+ aus G ist selbstverständlich nur ein Teil der Tätigkeiten, die im Zusammenhang mit der Strukturierung von Daten für ein relationales Datenmodell stehen. Aus diesem Grund wird im dritten Kapitel zunächst eine allgemeine Vorgehensweise für die logische Datenorganisation beschrieben. Zum einen um zu zeigen, welche - nicht rechnerunterstützten - Methoden zur Anwendung kommen können; zum anderen um zu zeigen, an welcher Stelle der Analyse die neu entwickelte Methode zum Einsatz kommt. Nachfolgend wird vorausgesetzt, daß alle unterschiedlich benannten Attribute unterschiedlich sind:
$A \neq B \neq C$. Ist nur die funktionale Abhängigkeit $A \rightarrow B$ angegeben, so gilt $B \not\rightarrow A$, andernfalls muß $A \rightarrow B$ und $B \rightarrow A$ angegeben werden. Sind jedoch die funktionalen Abhängigkeiten $A \rightarrow B \rightarrow C$ angegeben, dann gilt implizit $A \rightarrow C$.

2. UNTERSUCHUNG IN DER LITERATUR BEHANDELTER METHODEN

2.1 ANSÄTZE, DIE AUF EINER DATENVERWENDUNGSMATRIX AUFBAUEN

2.1.1 ALLGEMEINE Vorgehensweise

Eine Datenverwendungsmatrix, wie sie hier betrachtet werden soll, enthält in ihren Zeilen Datenelemente und in ihren Spalten die Verwendung der Datenelemente. Als Beispiel für die im Kapitel 2.1.2 verwendete Datenverwendungsmatrix A wäre folgendes denkbar:

	Rechnung	Kunden-statistik	Vertreter-statistik	Adreß-verzeichn.
Lieferant	0	0	0	1
Kunde	1	1	1	0
Auftragsnr.	0	0	1	0
Artikelnr.	1	1	0	0
Vertreter	0	0	1	0

Kommt ein bestimmtes Datenelement in einem Anwendungsfall vor, so wird das Matrixelement mit 1 gekennzeichnet. Kommt das Datenelement nicht vor, so hat das Matrixelement den Wert 0. Darüberhinaus besteht die Möglichkeit die Bedeutung der einzelnen Datenelemente in bestimmten Anwendungsfällen dadurch hervorzuheben, daß man statt der Ziffer 1 eine Gewichtungskennziffer z.B. 1 bis 9 vergibt. Dadurch wäre es beispielsweise möglich Schlüsselattribute von Nichtschlüsselattributen zu unterscheiden.
Um Datenmaterial zu gruppieren, das in Form einer solchen Datenverwendungsmatrix vorliegt, wäre ein denkbarer Weg die Clusteranalyse, die folgendermaßen vorgeht: [14)]

---

14) vgl. Späth, H.: Partitionierende Clusteranalyse für große Objektmengen mit binären Merkmalen am Beispiel von Firmen und deren Berufsgruppenbedarf. In: Fallstudien zur Clusteranalyse, hrsg. von H. Späth. München 1977, S.64

- Zunächst wird eine symmetrische Abstands- oder Ähnlichkeitsmatrix ermittelt. Dafür muß ein Abstands- / Ähnlichkeitsmaß gewählt werden, das die charakteristischen Eigenschaften der zu analysierenden Datenverwendungsmatrix möglichst gut wiedergibt. In den nachfolgenden Beispielen werden einige Abstands- bzw. Ähnlichkeitsmaße beschrieben und deren Auswirkungen auf das Ergebnis der Clusteranalyse gezeigt.

- Aufgrund dieser Abstands- / Ähnlichkeitsmatrix werden Cluster durch

  graphentheoretische,
  partitionierende oder
  hierarchische

  Verfahren ermittelt.

Als Beispiel für graphentheoretische Clusteranalyse wird nachfolgend das Verfahren von Freudenthaler und Maier vorgestellt. Als partitionierender Algorithmus wurde das Austauschverfahren von Späth ausgewählt.

Soll ein Clusteranalyseverfahren zur Unterstützung beim Aufbau der logischen Datenstruktur dienen, so muß es mindestens folgende Anforderungen erfüllen:

- gleiche Zeilen/Spalten müssen in einem Cluster zusammengefaßt werden.
- Zeilen/Spalten, die in keinem (von 0 verschiedenen) Matrixelement übereinstimmen, dürfen nicht zusammengefaßt werden.
- keine Zeile/Spalte darf mehreren Clustern zugeordnet werden.

Diese Anforderungen klingen zunächst trivial. Es wird aber noch gezeigt werden, daß sogar die ersten beiden Anforderungen nicht von allen Verfahren gewährleistet werden.

2.1.2 GRAPHENTHEORETISCHER ANSATZ DER CLUSTERANALYSE NACH FREUDENTHALER UND MAIER [15]

2.1.2.1 Zielsetzung

- Nach der Klassifikation sollen ähnliche Zeilen und/oder Spalten nebeneinander liegen.
- Keine Zeile/Spalte soll aus der Klassifikation ausgeschlossen werden.
- Keine Zeile/Spalte darf in mehreren Klassen vorkommen.
- Eine Klasse wird aufgrund eines noch zu bestimmenden Ähnlichkeitsmaßes w_x (Klassifikationsniveau) gebildet. Dieses Klassifikationsniveau darf bei einem paarweisen Vergleich aller Zeilen/Spalten einer Klasse nicht unterschritten werden.

---

15) vgl. Freudenthaler, B., Maier, B.: Die Entwicklung eines Datenmodells aus Informationsbedarfsanalysen - ein Klassifikationsproblem. Angewandte Informatik 12/77, S.522-532

2.1.2.2 Vorgehensweise

2.1.2.2.1 Bestimmung der Ähnlichkeitsmatrix W

Die Vorgehensweise soll anhand der bei Freudenthaler und Maier verwendeten "Matrix A" erläutert werden.
Es handelt sich dabei um eine überschaubare Datenverwendungsmatrix mit fünf Zeilen und vier Spalten.

i \ j	1	2	3	4
1	0	0	0	1
2	1	1	1	0
3	0	0	1	0
4	1	1	0	0
5	0	0	1	0

Bild 1: Matrix A

Diese Matrix ist insoweit besonders gut für den Vergleich der einzelnen Verfahren geeignet, weil sie zwei identische Zeilen (3 und 5) und zwei identische Spalten (1 und 2) enthält. Die Zeile 1 stimmt in keinem 1-Element mit einer anderen Zeile überein, dasselbe gilt für die Spalte 4.
Diese Datenverwendungsmatrix wird auch nachfolgend im Kapitel 2.1 immer als "Matrix A" benannt.
Hier soll die zeilenweise Klassifikation der Matrix A, die in den Zeilen die Datenelemente und in den Spalten die Verwendungen oder Kombinationsbedarfsprofile enthält,

beschrieben werden.
Für eine spaltenweise Klassifikation ist die Datenverwendungsmatrix zu transponieren.

Der Vergleich zweier Zeilen erfolgt mit dem Ähnlichkeitsmaß (Platzübereinstimmungsmaß):

$$w_{i,j} = m - \sum_{q=1}^{m} \left| a_{i,q} - a_{j,q} \right| \qquad 1=i,j=n$$

n: Anzahl der Zeilen
m: Anzahl der Spalten
$a_{i,j}$: Element der Datenverwendungsmatrix
$w_{i,j}$: Element der Ähnlichkeitsmatrix

i \ j	1	2	3	4	5
1	4	0	2	1	2
2	0	4	2	3	2
3	2	2	4	1	4
4	1	3	1	4	1
5	2	2	4	1	4

Bild 2: Ähnlichkeitsmatrix W

Die maximale Ausprägung des Ähnlichkeitsmaßes ist gleich der Anzahl der Spalten der Datenverwendungsmatrix und bestimmt damit die Werte der Hauptdiagonalen.
Als nachfolgend zu bestimmendes Klassifikationsniveau wird ein auszuwählender Ähnlichkeitswert verstanden, der bei keinem Vergleich unterschritten wird.

2.1.2.2.2 Bestimmung des bestmöglichen Klassifikationsniveaus

Als bestmögliches Klassifikationsniveau wird diejenige Ausprägung des Ähnlichkeitsmaßes gewählt, die möglichst groß ist und dennoch in allen Zeilen vorkommt.
Dieser Wert muß nicht mit dem am häufigsten auftretenden Ähnlichkeitsmaß identisch sein.
Der Benutzer kann im Verfahren von Freudenthaler und Maier anstelle des maschinell ermittelten "bestmöglichen Klassifikationsniveaus" ein von ihm gewähltes Klassifikationsniveau vorgeben. Um diese Auswahl des Klassifikationsniveaus zu erleichtern werden ihm alle Ausprägungen des Ähnlichkeitsmaßes und deren Häufigkeiten mitgeteilt.

Mit dem maschinell bestimmten oder frei gewählten Klassifikationsniveau wird eine neue Ähnlichkeitsmatrix ermittelt. Alle Matrixelemente der bisherigen Ähnlichkeitsmatrix, die kleiner sind als das Klassifikationsniveau w_x werden null (d.h. die zugehörigen Zeilen werden als nicht ähnlich gewertet); alle anderen Matrixelemente werden 1 (d.h. die zugehörigen Zeilen sind ähnlich).

i \ j	1	2	3	4	5
1	1	0	1	0	1
2	0	1	1	1	1
3	1	1	1	0	1
4	0	1	0	1	0
5	1	1	1	0	1

Bild 3: Ähnlichkeitsmatrix unter Berücksichtigung des Klassifikationsniveaus $w_x=2$

2.1.2.2.3 Klassifikation der Matrix A durch Subgraphenbildung

Mit der neuen Ähnlichkeitsmatrix (Bild 3) kann ein Graph gezeichnet werden, dessen Knoten die zu klassifizierenden Elemente der Datenverwendungsmatrix sind. Diese Knoten werden dann durch Kanten verbunden, wenn das zugehörige w_{ij} in der Ähnlichkeitsmatrix (Bild 3) gleich 1 ist, die zu klassifizierenden Elemente also ähnlich sind.

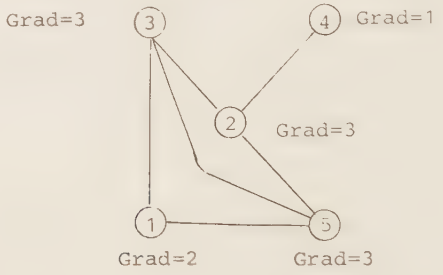


Bild 4: Graph für w_x=2

Der Grad der Knoten - als Anzahl der Kanten, die von einem Knoten hin- bzw. wegführen - dient hier als Maß um möglichst viele ähnliche Elemente zusammenzufassen. Im Beispiel haben die Knoten 2,3 und 5 denselben Grad. Für die Zeilen 2,3 und 5 gibt es also gleichviele ähnliche Zeilen. Deshalb wird derjenige Knoten gewählt, dessen einstufig erreichbare Knoten insgesamt den höchsten Grad haben.

Beispiel:

Knoten	Grad	Summe des Grads der einstufig erreichbaren Knoten
1	2	6
2	3	7
3	3	8
4	1	3
5	3	8

So sind beispielsweise vom Knoten 3 aus die Knoten 1,2 und 5 einstufig erreichbar. Dann ergibt sich die Summe aus dem Grad der Knoten 1,2 und 5; also 2+3+3=8.

Als Subgraphenzentrumsknoten (SZK) wird der Knoten gewählt, der den höchsten Grad hat und dessen benachbarte Knoten gleichzeitig den höchsten Grad (die größtmögliche Ähnlichkeit) haben.
Im vorausgehenden Beispiel wäre das der Knoten 3 oder 5. Ausgehend von diesem Subgraphenzentrumsknoten (SZK) werden alle einstufig erreichbaren Knoten zu einem Subgraphen zusammengefaßt.
Für den SZK 3 wird der Subgraph 3-1-2-5 gebildet.
Da der Knoten 4 noch in keinem Subgraphen enthalten ist, muß noch der Subgraph 4-2 gebildet werden.

In diesem ersten Schritt der Subgraphenbildung wird lediglich festgelegt, daß es beispielsweise eine einstufige Verbidung von Knoten 3 zum Knoten 1 gibt und eine Verbindung vom Knoten 3 zum Knoten 2 oder 5, nicht jedoch ob es eine Verbindung vom Knoten 1 zum Knoten 2 gibt. Die Überprüfung mehrstufiger Verbindungen erfolgt in der sogenannten "Baumbildung". Ausgehend von der einstufigen Verbindung 3-1 wird geprüft, ob für den Knoten 1 auch Verbindung zu den Knoten 2 oder 5 besteht.
Für den Knoten 5 ist dies der Fall, nicht jedoch für den Knoten 2. Demnach kann ein Subgraph 3-1-5 gebildet werden.

Aus der Bildung vollständiger Subgraphen ergeben sich drei Klassen:

Klasse	Subgraph
1	3 - 1 - 5
2	3 - 2 - 5
3	4 - 2

2.1.2.2.4 Bestimmung disjunkter Metaklassen

Da die Forderung besteht, daß kein Element in mehreren Klassen vorkommen darf, scheidet die Klasse 2 aus.
Das Ergebnis ist somit ein Cluster (Klasse 1) mit den Elementen 3-1-5 und ein Cluster (Klasse 3) mit den Elementen 4-2.

i \ j	1	2	3	4
3	0	0	1	0
1	0	0	0	1
5	0	0	1	0
4	1	1	0	0
2	1	1	1	0

Bild 5: Matrix A zeilenklassifiziert

2.1.2.2.5 Spaltenweise Klassifikation

Für die spaltenweise Klassifikation wird die Datenverwendungsmatrix transponiert und der oben beschriebene Vorgang wiederholt.

i \ j	1	2	3	4	5
1	0	1	0	1	0
2	0	1	0	1	0
3	0	1	1	0	1
4	1	0	0	0	0

Bild 6: transponierte Matrix A

i \ j	1	2	3	4
1	5	5	2	2
2	5	5	2	2
3	2	2	5	1
4	2	2	1	5

Bild 7: Ähnlichkeitsmatrix W

i \ j	1	2	3	4
1	1	1	1	1
2	1	1	1	1
3	1	1	1	0
4	1	1	0	1

Bild 8: Ähnlichkeitsmatrix unter Berücksichtigung des Klassifikationsniveaus $w_x=2$

Knoten	Grad	Summe des Grads der einstufig erreichbaren Knoten
1	3	7
2	3	7
3	2	6
4	2	6

Als Subgraphenzentrumsknoten wird der Knoten 1 ausgewählt. In diesem Fall ergibt sich ein Subgraph der alle Knoten beinhaltet: 1 - 2 - 3 - 4.
Bei der Baumbildung (mehrstufige Verbindung der Knoten) entstehen wiederum drei Klassen:

Klasse	Subgraph
1	1 - 2 - 3
2	1 - 3
3	1 - 4

Aus diesem Beispiel ist bereits erkennbar, daß die Bildung disjunkter Metaklassen nicht ganz einfach ist. Da jede Zeile in einer Klasse nur einmal vorkommen darf können folgende Cluster gebildet werden:

Klasse 1: 1 - 2 - 3
2: 4

oder

Klasse 1: 1 - 2
2: 3
3: 4

i \ j	1	2	3	4
3	0	0	1	0
1	0	0	0	1
5	0	0	1	0
4	1	1	0	0
2	1	1	1	0

Bild 9: Matrix A zeilen- und spaltenklassifiziert

2.1.2.3 Zusammenfassung

Das Verfahren von Freudenthaler und Maier liefert nicht eine "bestmögliche" Klassenbildung sondern es zeigt möglichst viele verschiedene Klassenbildungsmöglichkeiten auf.
Das erleichtert es dem Benutzer nicht unbedingt, zu einer für ihn günstigen Clusterbildung zu gelangen. Dieses Problem ist bereits aus der spaltenweisen Klassifikation der Matrix A erkennbar. Schwierigkeiten traten immer dann auf, wenn ein Element nicht maschinell einer bestimmten Metaklasse zugeordnet werden konnte.
Der Algorithmus kann - wie bereits erwähnt - beliebig oft mit - vom Benutzer vorgegebenen - Klassifikationsniveaus durchlaufen werden. Die Vielzahl der Ergebnisse ist, wie später noch gezeigt werden wird, insbesondere bei größeren Matrizen eher verwirrend.

2.1.3 PARTITIONIERENDER ANSATZ DER CLUSTERANALYSE NACH SPÄTH [16]

2.1.3.1 Zielsetzung

Die Anzahl der Klassen, in die die Datenverwendungsmatrix unterteilt werden soll, wird hier zwar nicht fest vorgegeben, kann sich aber zwischen einem vorzugebenden minimalen und maximalen Wert bewegen. Die Schrittweite, in der die Klassenanzahl variiert werden soll, kann ebenfalls vorgegeben werden.
Für jede Klasse wird ein Homogenitätsmaß e_k ermittelt, dessen Summe über alle Klassen minimiert werden soll. Dabei handelt es sich um ein lokales Optimum, das globale Optimum wäre nur mit wesentlich höherer Rechenzeit zu ermitteln.
Da dieser Algorithmus nicht unabhängig von der gewählten Anfangspartition ist, soll jedes Anfangscluster möglichst die gleiche Anzahl von Elementen haben.
Dies wird durch einen Pseudozufallszahlengenerator erreicht.
Um den Speicherplatzbedarf möglichst gering zu halten, werden bei der symmetrischen Abstandsmatrix nur die Werte oberhalb der Hauptdiagonalen gespeichert.

Das Verfahren wird nachfolgend am Beispiel der Matrix A erläutert:

$$A = (a_{ij}) \qquad \begin{matrix} 1 \leq i \leq n \\ 1 \leq j \leq m \end{matrix}$$

---

16) vgl. Späth, H.: Partitionierende Cluster-Analyse für größere Objektmengen mit binären Merkmalen am Beispiel von Firmen und deren Berufsgruppenbedarf. In: Fallstudien zur Clusteranalyse, hrsg. von H. Späth. München 1977, S.63-80

Minimiert werden soll beim Austauschverfahren die Summe der Einzelhomogenitäten e_k. Also die Summe der - mit der reziproken Klassenanzahl - gewichteten Summen aller einfach gerechneten Abstände. [17)]

$$D\ (C_1 \ldots\ldots C_l) = \sum_{k=1}^{l} e_k = \sum_{k=1}^{l} \left|\frac{1}{C_k}\right| \sum_{i \in C_k} \sum_{\substack{j \in C_k \\ j > i}} t_{ij} \dashrightarrow \min$$

mit

t_{ij} = Abstandsmaß

$C_1 .. C_l$ = gesuchte Partitionen

$\frac{1}{C_k}$ = Anzahl der Elemente des k-ten Clusters

Das Austauschverfahren wird schrittweise solange verbessert, bis man ein lokales Optimum erreicht hat. Dabei wird eine Zeile aus ihrem Cluster entfernt und versuchsweise einem anderen Cluster zugeordnet. Verringert sich dadurch der Zielfunktionswert D, so bleibt die Zeile dem neuen Cluster zugeordnet, andernfalls wird diese Zuordnung wieder rückgängig gemacht. [18)]

---

17) vgl. Späth, H.: Partitionierende Cluster-Analyse bei Binärdaten am Beispiel von bundesdeutschen Hochschulen und Diplomstudiengängen. Zeitschrift Operations Research, 21 (1977), S.B87

18) vgl. Späth, H.: Numerische Erfahrungen zu heuristischen Lösungsverfahren beim Varianzkriterium in der Cluster-Analyse. Angewandte Informatik, 2/77, S.68
und Späth, H.: Partitionierende Cluster-Analyse für größere Objektmengen mit binären Merkmalen am Beispiel von Firmen und deren Berufsgruppenbedarf, a.a.O., S. 65 f

2.1.3.2 Vorgehensweise

2.1.3.2.1 Bestimmung der Abstandsmatrix

2.1.3.2.1.1 Abstandsmatrix mit dem TANIMOTO-Abstand

Da die Zielfunktion minimiert wird, wird vom Abstands-, nicht vom Ähnlichkeitsmaß ausgegangen.
Als Abstandsmaß wurde zunächst der Tanimoto-Abstand[19] gewählt:

$$t_{ij} = \frac{z_i + z_j - 2v_{ij}}{z_i + z_j - v_{ij}} \qquad 1 \leq i,j \leq n$$

mit der Zeilensumme der Datenverwendungsmatrix

$$z_p = \sum_{q=1}^{m} a_{pq} \qquad p = i,j$$

und $$v_{ij} = \sum_{q=1}^{m} \min (a_{iq}, a_{jq})$$

Das Tanimoto-Abstandsmaß kann Werte von 0 bis 1 annehmen. Zwei Zeilen der Datenverwendungsmatrix sind dann ähnlich, wenn der Abstand 0 ist und nicht ähnlich, wenn der Abstand 1 ist.
Für die Datenverwendungsmatrix A :

i \ j	1	2	3	4
1	0	0	0	1
2	1	1	1	0
3	0	0	1	0
4	1	1	0	0
5	0	0	1	0

19) vgl. Späth, H.: Cluster-Analyse-Algorithmen zur Objektklassifizierung und Datenreduktion. München 1975, S.30
und Späth, H.: Partitionierende Cluster-Analyse für größere Objektmengen, a.a.O., S. 66

ergibt sich

$V = (v_{ij})$

	1	2	3	4	5
1	1	1	1	1	1
2	1	1	1	1	1
3	1	1	1	1	1
4	1	1	1	1	1
5	1	1	1	1	1

Die Werte der Hauptdiagonalen entsprechen der Zeilensumme z_z der Matrix A.
Daraus erhält man die symmetrische TANIMOTO-Abstandsmatrix

$T = (t_{ij})$

	1	2	3	4	5	Zeilensumme:
1	0	0	0	0	0	12/3
2	0	0	2/3	1/3	2/3	8/3
3	0	2/3	0	0	0	8/3
4	0	1/3	0	0	0	10/3
5	0	2/3	0	0	0	8/3

Gesamtsumme: 46/3

Da in das Abstandsmaß t_{ij} nur die Zeilensumme der Datenverwendungsmatrix, nicht aber die Anzahl der 0-Elemente eingeht, werden nur die 1-Elemente berücksichtigt.
D.h. nur das Vorhandensein von Datenelementen in bestimmten Verwendungen wird berücksichtigt, nicht jedoch das Nichtvorhandensein.
Statt des TANIMOTO-Abstands kann auch ein anderes Abstandsmaß verwendet werden, z.B. der Matching-Koeffizient.

2.1.3.2.1.2 Abstandsmatrix mit dem Matching-Koeffizienten

Der TANIMOTO-ABSTAND liefert Werte, die alle zwischen 0 und 1 liegen. Soll dagegen mit einer breiteren Skalierung und mit ganzzahligen Werten gearbeitet werden, so bietet sich der Matching-Koeffizient an:[20)]

$$t_{ij} = \sum_{q=1}^{m} \left| a_{iq} - a_{jq} \right| = z_i + z_j - 2v_{ij}$$

Dieses Abstandsmaß liefert ganzzahlige Werte von 0 bis m. Es ist das korrespondierende Abstandsmaß zu dem Ähnlichkeitsmaß, das im Verfahren von Freudenthaler und Maier verwendet wird. Dort wird mit dem Ähnlichkeitsmaß:

$$w_{ij} = m - \sum_{q=1}^{m} \left| a_{iq} - a_{jq} \right| = m - t_{ij}$$

gearbeitet.
Man erhält mit dem Matching-Koeffizienten:

$T = (t_{ij})$

i \ j	1	2	3	4	5	Zeilensumme:
1	0	4	2	3	2	11
2	4	0	2	1	2	9
3	2	2	0	3	0	7
4	3	1	3	0	3	10
5	2	2	0	3	0	7
						Gesamtsumme: 44

Bild 10: Abstandsmatrix mit dem Matching-Koeffizienten

Der Matching-Koeffizient liefert für die Matrix A differenzierte Ergebnisse als der TANIMOTO-Abstand.

---

20) vgl. Späth, H.: Partitionierende Cluster-Analyse für große Objektmengen mit binären Merkmalen, a.a.O., S.78

So haben beispielsweise die Zeilen 1 und 2 sowie 1 und 3 unterschiedliche Abstände ($t_{12}=4$, $t_{13}=2$) während beim TANIMOTO-Abstand beide Abstände gleich 1 waren. Beim TANIMOTO-Abstand wurden also diese Zeilen generell als "nicht ähnlich" klassifiziert.

2.1.3.2.2 Ermittlung des Homogenitätsmaßes

Für jedes Cluster wird - ausgehend von einem zufällig gewählten Anfangscluster - ein Homogenitätsmaß

$$e_k = \left|\overline{C_k}\right| \sum_{i \in C_k} \sum_{j \in C_k} t_{ij} \qquad j > i$$

gebildet, wobei $|C_k|$ die Anzahl der Elemente im k-ten Cluster ist. Es handelt sich somit um die durchschnittliche Summe aller Abstände dieses Clusters.

2.1.3.2.3 Optimierung durch das Austauschverfahren

Aus den zufällig gewählten Ausgangspartitionen werden neue Cluster gebildet, indem man aus einem Cluster ein Element entfernt und dieses Element einem anderen Cluster hinzufügt. Wird der Zielfunktionswert dadurch kleiner, so bleibt das Element dem neuen Cluster zugeordnet.

$$C_{k\ neu} = C_k - C_i \qquad C_{p\ neu} = C_p + C_i$$

Dabei ist C_i eine nichtleere, aber tatsächlich in C_k enthaltene Teilmenge und es gilt $C_k \neq 0$ mit $C_k \neq C_i$ und $C_i \in C_k$ und hier $C_i = \{i\}$ also einelementig ist. [21] Außerdem gilt, daß $C_{k\ neu}$ eine nichtleere Differenzmenge von C_k und C_i ist.
Da in der Zielfunktion die Summe der Homogenitätsmaße minimiert werden soll, bleibt die neue Klassenzuordnung nur dann bestehen, wenn

$$e_k + e_p > e_{k\ neu} + e_{p\ neu}$$

es muß also ein $p \neq k$ bestehen, für das gilt:

$$f(p) = e_k + e_p - e_{k\ neu} - e_{p\ neu} > 0$$

Dieses Austauschverfahren wird für alle Cluster und innerhalb der Cluster für alle Elemente vorgenommen und zwar solange, bis der Zielfunktionswert nicht mehr verkleinert werden kann.
Die Anzahl der zu bildenden Klassen können dabei von einem minimalen bis zu einem maximalen Wert variiert werden.

---

21) vgl. Späth. H.: Cluster-Analyse Algorithmen, a.a.O., S.68f

2.1.3.3 Ergebnis des Austauschverfahrens am Beispiel der Matrix A

Clusterbildung mit den Zeilen	bezogen auf die rechte obere Dreieckshälfte der Abstandsmatrix Zielfunktionswert mit dem TANIMOTO-Abstand	Matching-Koeff.	bezogen auf die vollständige Abstandsmatrix Zielfunktionswert mit dem TANIMOTO -Abstand	Matching-Koeff.
1,2,3 4,5	2,72	7,83	6,11	17,5
1,3 2,4,5	3,39	9,67	6,22	17,67
1,3,5 2,4	3,00	8,67	6,11	17,83
1,4 2,3,5	3,39	9,67	6,34	18,17
1,2,4 3,5	2,72	7,83	6,00	17,00
1 2,4 3,5	5,84	16,5	9,67	27,5

Stellvertretend für alle Möglichkeiten der Clusterbildung, die für die Matrix A bestehen, wurden sechs Beispiele (vgl. S. 31) berechnet.
Verwendet man zur Bestimmung des Zielfunktionswertes - wie bei Späth angegeben - nur die rechte obere Dreieckshälfte der Abstandsmatrix, so kann man anhand des minimalen Zielfunktionswertes (2,72) nicht erkennen ob die Möglichkeit 1 oder 5 der Clusterbildung günstiger ist. Da die Forderung gestellt wurde, nicht nur ähnliche sondern vor allem gleiche Zeilen zusammenzufassen, müssen die Zeilen 3 und 5 demselben Cluster zugeordnet werden. Dies trifft nur auf die 5. Möglichkeit der Clusterbildung zu.
Verwendet man jedoch die vollständige Abstandsmatrix zur Bestimmung der Zielfunktion, so erhält man sowohl mit dem TANIMOTO-Abstand als auch mit dem Matching-Koffizienten für das fünfte Beispiel die günstigste Lösung, also für die Clusterbildung mit den Zeilen 1,2,4 und 3,5. Da die Zeile 1 in keinem 1-Element mit den Zeilen 2 oder 4 übereinstimmt, wäre es günstiger diese Zeile einem eigenen Cluster zuzuordnen.
Die Zielfunktionswerte, die sich bei unterschiedlicher Anzahl der zu bildenden Cluster ergeben, sind jedoch nicht vergleichbar, wie aus dem sechsten Beispiel (vgl. S.31) ersichtlich ist.

2.1.4 BETRACHTUNG WEITERER ÄHNLICHKEITSMAßE[22)]

2.1.4.1 S-Koeffizient

Da die bisherigen Ähnlichkeitsmaße nicht zu voll zufriedenstellenden Ergebnissen geführt haben, werden hier noch weitere Ähnlichkeitsmaße betrachtet, die eine Gewichtung der Zeilen bzw. der Spalten ermöglichen. Der S-Koeffizient definiert die Ähnlichkeit durch das Verhältnis der Anzahl der in beiden Varianten gemeinsam enthaltenen Merkmale zur Anzahl der in beiden Varianten insgesamt enthaltenen Merkmale bzw. den Gewichtssummen. Bei den in diesem Abschnitt aufgeführten Ähnlichkeitsmaßen werden folgende Symbole verwendet:

w_{ij} = Ähnlichkeit zwischen den Elementen i und j

d_{ij} = Anzahl der übereinstimmenden 1-Merkmale bzw. Summe der Gewichte G_i der übereinstimmenden 1-Merkmale

b_{ij} = Anzahl bzw. Summe der Gewichte G_i der Merkmale für die gilt: $a_{iq} = 0$ und $a_{jq} = 1$

c_{ij} = Anzahl bzw. Summe der Gewichte G_i der Merkmale für die gilt: $a_{iq} = 1$ und $a_{jq} = 0$

Damit lautet die Formel für den S-Koeffizienten:

$$w_{ij} = \frac{d_{ij}}{b_{ij} + c_{ij} + d_{ij}}$$

22) vgl. Zimmermann, G.: Typenbildung mit Methoden der Clusteranalyse. Fortschrittliche Betriebsführung, 26 (1977), S.384

Ergebnis am Beispiel der Matrix A:[23)]

i \ j	1	2	3	4
1	0	0	0	1
2	1	1	1	0
3	0	0	1	0
4	1	1	0	0
5	0	0	1	0

$1 \leq i \leq n$

$1 \leq j \leq m$

Bild 11: Matrix A

i \ j	1	2	3	4	5
1	1.00	0.00	0.00	0.00	0.00
2	0.00	1.00	0.32	0.66	0.32
3	0.00	0.32	1.00	0.00	1.00
4	0.00	0.66	0.00	1.00	0.00
5	0.00	0.32	1.00	0.00	1.00

$1 \leq i,j \leq n$

Bild 12: Ähnlichkeitsmatrix mit dem S-Koeffizienten

i \ j	1	2	3	4
1	1.00	1.00	0.25	0.00
2	1.00	1.00	0.25	0.00
3	0.25	0.25	1.00	0.00
4	0.00	0.00	0.00	1.00

$1 \leq i,j \leq m$

Bild 13: Ähnlichkeitsmatrix für die transponierte Matrix A mit dem S-Koeffizienten

Auffallend ist, daß in dieser Ähnlichkeitsmatrix Elemente null sind, die in der Ähnlichkeitsmatrix nach dem Verfahren von Freudenthaler und Maier noch von Null verschieden sind. Hier werden diese Elemente also bereits nicht mehr als ähnlich gewertet.

23) Die Ergebnisse wurden nicht gerundet.
Es wurde mit einer Gewichtung von $G_i = 1$ gerechnet.

2.1.4.2 Erste Variante des S-Koeffizienten

Berücksichtigt man beim Vergleich von zwei nicht übereinstimmenden Merkmalen (0 und 1 bzw. 1 und 0) jeweils nur den kleineren Wert, so gilt:

$$w_{ij} = \frac{d_{ij}}{d_{ij} + \min (c_{ij}, b_{ij})}$$

Man erhält dann für die Matrix A folgende Ähnlichkeitsmatrix:

i \ j	1	2	3	4	5
1	1.00	0.00	0.00	0.00	0.00
2	0.00	1.00	1.00	1.00	1.00
3	0.00	1.00	1.00	0.00	1.00
4	0.00	1.00	0.00	1.00	0.00
5	0.00	1.00	1.00	0.00	1.00

und für die transponierte Matrix A die Ähnlichkeitsmatrix:

i \ j	1	2	3	4
1	1.00	1.00	0.50	0.00
2	1.00	1.00	0.50	0.00
3	0.50	0.50	1.00	0.00
4	0.00	0.00	0.00	1.00

Man erkennt sofort, daß hier die Ähnlichkeiten zu wenig differenziert werden. Die Datenelemente 2 und 3, 2 und 4, 2 und 5 (Zeilen der Matrix A) scheinen ähnlich bzw. identisch zu sein, obwohl das nicht der Fall ist.

2.1.4.3 Zweite Variante des S-Koeffizienten

Verwendet man statt dem min (c_{ij}, b_{ij}) den größten Wert aus diesen beiden Komponenten, so erhält man:

$$w_{ij} = \frac{d_{ij}}{d_{ij} + \max (c_{ij}, b_{ij})}$$

und damit für die Matrix A die Ähnlichkeitsmatrix:

i \ j	1	2	3	4	5
1	1.00	0.00	0.00	0.00	0.00
2	0.00	1.00	0.32	0.66	0.32
3	0.00	0.32	1.00	0.00	1.00
4	0.00	0.66	0.00	1.00	0.00
5	0.00	0.32	1.00	0.00	1.00

und die Ähnlichkeitsmatrix für die transponierte Matrix A:

i \ j	1	2	3	4
1	1.00	1.00	0.32	0.00
2	1.00	1.00	0.32	0.00
3	0.32	0.32	1.00	0.00
4	0.00	0.00	0.00	1.00

Die Ergebnisse sind in diesem Fall mit denen des S-Koeffizienten identisch. Diese Variante des S-Koeffizienten ist nur dann von Bedeutung, wenn die Datenelemente unterschiedlich gewichtet werden.

2.1.4.4 Dritte Variante des S-Koeffizienten

Bei dieser Variante wird die Übereinstimmung der 1-Elemente geringer gewertet. Da es bei der Analyse einer Datenverwendungsmatrix aber gerade auf diese Übereinstimmung ankommt, ist dieses Ähnlichkeitsmaß weniger geeignet. Es sei hier nur der Vollständigkeit halber aufgeführt.

$$w_{ij} = \frac{d_{ij}}{2d_{ij} + b_{ij} + c_{ij}}$$

Daraus ergeben sich für die Matrix A in der Ähnlichkeitsmatrix folgende Werte:

i \ j	1	2	3	4	5
1	0.50	0.00	0.00	0.00	0.00
2	0.00	0.50	0.25	0.39	0.25
3	0.00	0.25	0.50	0.00	0.50
4	0.00	0.39	0.00	0.50	0.00
5	0.00	0.25	0.50	0.00	0.50

und als Ähnlichkeitsmatrix für die transponierte Matrix A:

i \ j	1	2	3	4
1	0.50	0.50	0.19	0.00
2	0.50	0.50	0.19	0.00
3	0.19	0.19	0.50	0.00
4	0.00	0.00	0.00	0.50

2.1.4.5 Zusammenfassung

Die im Kapitel 2.1.4 gezeigten Ähnlichkeitsmaße haben - wie der TANIMOTO-Abstand - den Nachteil, daß sie nicht ganzzahlig sind und bei einer Gewichtung von 1 Werte zwischen 0 und 1 oder sogar nur zwischen 0 und 0,5 annehmen.
Sie gewinnen allerdings dann an Bedeutung, wenn es darauf ankommt, auch solche Elemente in die Ähnlichkeit einzubeziehen, bei denen das eine Merkmal 0 und das andere - zu vergleichende - Merkmal 1 ist oder umgekehrt.

2.1.5 CLUSTERBILDUNG UNTER BERÜCKSICHTIGUNG DER ZUGRIFFS-ANFORDERUNGEN UND NORMALISIERUNG NACH DÖRINGER [24)]

2.1.5.1 Zielsetzung

Ausgangspunkt ist - ähnlich, wie bisher - eine Datenverwendungsmatrix, deren Elemente jedoch nicht null oder eins sind, sondern statt der Kennzeichnung 1 zusätzlich Aussagen über die Zugriffsart der Datenelemente machen und darüber, ob es sich bei dem Datenelement um ein identifizierendes Feld handelt oder nicht. Dadurch soll bei der Strukturierung der Datenelemente von vornherein die Zugriffsart berücksichtigt werden. Die so gefundenen Relationen werden anschließend normalisiert.

---

24) vgl. Döringer, H.: Eine Methodik zur Strukturierung von Datenbanken. Angewandte Informatik, 1978, Heft 3, S.93-103

2.1.5.2 Vorgehensweise

2.1.5.2.1 Festlegung der Zugriffs- und Updateanforderungen

Die Anforderungen (Datenverwendungen) werden gegenüber den bisherigen Beispielen in reine Abfragen (Anforderungsart Z) und reine Änderungsdienste (Anforderungsart U) unterteilt.[25)]

Anforderungsnummer / Anforderungsart / Feld	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	U	U	U	U
Auftragsnummer	o	oK	oK	o	o	oK	oK	oK	oK			oK	oK		
Datum						x						x			
Termin		x				x						x			
Positionsnr.	o		o		o		o		o				oK		
Mengeneinheit			x		x		x						x		
Anzahl			x		x		x						x		
Kundennummer				K			x	x				x			
Kundenname								x				x			
Ort								x				x			
Straße								x				x			
Hausnummer								x				x			
Rabattklasse								x				x			
Teilenummer	K		x		K				x	oK	oK			oK	oK
Teilename									x	x	x				x
Listenpreis									x	x				x	
Einzelkosten															
Material										x				x	
Einzelkosten															
Lohn										x				x	
Rohmaterialnr.											x				x
Rohmat. Menge											x				x
Gewicht											x				x

Bild 14: Datenverwendungsmatrix nach Döringer[25)]

Die Datenelemente, die in einer bestimmten Anwendung vorkomme werden hier nicht mit 1 gekennzeichnet, sondern zusätzlich klassifiziert als

- o identifizierendes Feld
- K Suchargument
- x ohne besondere Funktion
- oK identifizierendes Feld, das als Suchargument dient

25) vgl. Döringer, H.: Eine Methodik zur Strukturierung von Datenbanken, a.a.O., S.94

2.1.5.2.2 Bildung von Relationen und deren Normalisierung

Zunächst werden diejenigen Anwendungen zusammengefaßt, die denselben Primärschlüssel verwenden und solche, die gleiche oder ähnliche Zugriffsarten haben.

Anforderungsnummer / Anforderungsart / Feld	I						II					III				I	II	III
	1	3	5	7	9	13	2	4	6	8	12	10	11	14	15			
	Z	Z	Z	Z	Z	U	Z	Z	Z	Z	U	Z	Z	U	U	Z/U	Z/U	Z/U
Auftragsnummer	o	oK	o	oK	oK	oK	oK	o	oK	oK	oK					oK	oK	
Datum									x		x						x	
Termin							x		x		x						x	
Positionsnr.	o	o	o	o	o	oK										oK		
Mengeneinheit		x	x	x		x										x		
Anzahl		x	x	x		x										x		
Kundennummer				x				K		x	x					x	K	
Kundenname										x	x						x	
Ort										x	x						x	
Straße										x	x						x	
Hausnummer										x	x						x	
Rabattklasse										x	x						x	
Teilenummer	K	x	K		x							oK	oK	oK	oK	K		oK
Teilename					x							x	x		x	x		x
Listenpreis					x							x		x		x		x
Einzelkosten																		
Material												x		x				x
Einzelkosten																		
Lohn												x		x				x
Rohmaterialnr.													x		x			x
Rohmat.Menge													x		x			x
Gewicht													x		x			x

Bild 15: Gruppierung der Anforderungen [26)]

Die so gebildeten Relationen werden in die dritte Normalform transformiert.

---

26) vgl. Döringer, H.: Eine Methodik zur Strukturierung von Datenbanken, a.a.O., S.95
und ADV-ORGA (Hrsg.): Seminarunterlagen Datenbankentwurfstechniken. ORGTECH 1. Wilhelmshaven 1980, S.3.5.370 ff

2.1.5.2.3 Zusammenfassung von Relationen der 3NF und Gliederung in Segmente

Diese Relationen in 3NF werden - sofern sie denselben Primärschlüssel haben - zusammengefaßt.
Dadurch kann eventuell vorhandene Redundanz entfernt werden, es gehen jedoch vorwiegend kleinere Relationen in größeren Relationen auf.
Um zu vermeiden, daß unnötige Daten - durch zu große Relationen bei Anwendungen "mitgeschleppt" werden, die dort gar nicht benötigt werden, wird für diese zusammengefaßten Relationen erneut eine Datenverwendungsmatrix erstellt.
Ausgehend von dieser Datenverwendungsmatrix wird eine Durchschnittsbildung der Datenelement-Mengen vorgenommen, um so die zu großen Relationen in kleinere Segmente zu gliedern.

2.1.5.3 Zusammenfassung

Dieses Verfahren ähnelt in seinem ersten Teil den Verfahren der Cluster-Analyse und im zweiten Teil den - noch zu beschreibenden - Verfahren der Relationensynthese (Kapitel 2.3).
Im Gegensatz zum Ansatz der Relationensynthese wird hier eine Vorauswahl über die zu erwartenden Zugriffsanforderungen getroffen.
Sowohl bei der Relationensynthese als auch im vorliegenden Verfahren wird eine Normalisierung der gefundenen Relationen in 3NF vorgenommen.
Es liegt dieser Normalisierung jedoch kein Algorithmus zugrunde und es wird kein Nachweis geführt, daß die so gefundenen Relationen immer in 3NF sein müssen.
Da von einer Datenverwendungsmatrix ausgegangen wird, müssen die Attribute in einer Relation nicht zwingend voll funktional abhängig vom Schlüsselattribut sein.
In den dargestellten Beispielen kann jedoch davon ausgegangen werden, daß voll funktionale Abhängigkeit besteht.
Durch die Durchschnittsbildung wird zwar Redundanz verringert, es geht aber aus der Beschreibung nicht hervor ob dadurch Relationen in optimaler 3NF erreicht werden können.

2.1.6 CLUSTERBILDUNG DURCH BERÜCKSICHTIGUNG DER REIHENFOLGE DER ORDNUNGSKRITERIEN NACH KOREIMANN [27]

2.1.6.1 Zielsetzung

Es wird eine generelle Systematik zur Strukturierung von Datenbanken aufgezeigt. Dabei werden folgende Ziele berücksichtigt:[27]

organisatorisch-technische Zielsetzungen

- optimale Betriebsablaufsteuerung
- geringe Zeiten für Generierung und Updating
- Sicherheit (Restart, Checkpoint, Logging)
- Verwaltung von Massendaten
- Trennung von Anwendungslogik und Datenlogik

anwendungsspezifische Zielsetzungen

- organisatorische Abstimmung der Anwendungen
- modularer Aufbau
- änderungs- und wartungsfreundliche Teilsysteme
- Datenschutz
- Integration mehrerer Anwendungen

benutzerspezifische Zielsetzung

- Direktabfragen
- große Variationsmöglichkeit bei Abfragen
- änderungsfreundliche Teilinformationssysteme
- Zugriffssicherung
- Ausbaufähigkeit des Gesamtsystems.

---

27) vgl. Koreimann, D.S.: Anwendungsanalyse - ein pragmatischer Ansatz für den Entwurf komplexer Datenbankorganisationen. Angewandte Informatik, 1974, S.1-5

2.1.6.2 Vorgehensweise

2.1.6.2.1 Anwendungsanalyse

Unter Anwendungsanalyse wird hier die Analyse aller aufeinander abgestimmten Aktivitäten verstanden, die darauf ausgerichtet sind,mereren Anwendungen Zugriff zur gemeinsamen Datenbank zu ermöglichen.
Ausgangspunkt ist wiederum eine Datenverwendungsmatrix, die sich von den bisher dargestellten Datenverwendungsmatrizen dadurch unterscheidet, daß sie eine unterschiedliche Gewichtung der Datenelemente vorsieht. Die Elemente der Datenmatrix nehmen nicht die Werte 0 oder 1 an, oder werden in Schlüssel- und Nichtschlüsselattribute unterschieden, sondern es werden alle Datenelemente in der Reihenfolge, in der sie als Ordnungskriterium in Frage kommen, durchnumeriert.
Dabei ist

1 = Hauptordnungskriterium
2 = 1. Nebenordnungskriterium
9 = 8. Nebenordnungskriterium

2.1.6.2.2 Redundanzanalyse

Die Redundanzanalyse soll hier die einzelnen Datenelemente auf formale, inhaltliche und syntaktische Richtigkeit und Plausibilität untersuchen.
So sollen beispielsweise Datennamen und deren Bedeutung vereinheitlicht werden, damit Doppelbezeichnungen vermieden werden. Darüberhinaus werden die Datenelemente vorab bereits zu Segmenten zusammengefaßt.
Das Ergebnis der Redundanzanalyse ist also ein "bereinigter" Datenkatalog, in dem bereits Gruppen von Datenelementen zusammengefaßt sind.

Bezeichnung d. entities	Segment-Name	Bedeutung des Segments
Name Adresse Schulbezirk Schulnr.	Schulname	Adresse und Kennzeichen der Schule

Bild 16: Beispiel eines Datenkatalogs

2.1.6.2.3 Bildung von Anwendungsfamilien

Unter Anwendungsfamilien werden hier Anwendungen verstanden, die durch organisatorische und datentechnische Ähnlichkeit zusammengefaßt werden können. Kriterien für die Bildung von Anwendungsfamilien sind z.B.:[28)]

organisatorisch begründete Kriterien
- zeitliche Abstimmung der Anwendungen
- ähnlicher oder gleicher Output
- ähnlicher oder gleicher Verarbeitungs-algorithmus

datentechnisch begründete Kriterien
- gleiche Ordnungskriterien
- Verwendung gleicher Daten oder Datengruppen
- datenerfassungs-orientierte Anwendung

systemtechnisch begründete Kriterien
- ähnliche Anforderungen für Sicherheitsvorkehrungen
- optimale Betriebsablaufsteuerung

---

28) vgl. Koreimann, D.S.: Anwendungsanalyse, a.a.O., S.2f
und Meller, Ch.: Eine Methode des informationsbedarfs-orientierten Datenbankentwurfs.
Angewandte Informatik, 1980, S. 226f

Anhand dieser Kriterien werden alle Anwendungen zu einer Anwendungsfamilie zusammengefaßt, die gleiche Datenelemente in gleicher oder ähnlicher Reihenfolge verarbeiten.

Anwendungen / Daten	Schulliste	Schulbelegung	Schulgattungsanalyse
Schulnummer	1	1	2
Schulbezirk	3	3	3
Schulgattung	2	2	1
Schuladresse	4	5	-
Schülernummer	-	-	4
Aktuel.Schule	-	-	4
Schülerkz.	-	-	5

Bild 17: Beispiel für Daten einer Anwendungsfamilie

2.1.6.2.4 Strukturierung der Datenbank

Hier wird sowohl auf die logische als auch auf die physische Strukturierung der Datenbank eingegangen. Allerdings werden keine Regeln aufgezeigt, wie man von den Anwendungsfamilien zu einer optimalen Struktur der Datenbank gelangt. Das aufgezeigte Beispiel orientiert sich am hierarchischen Datenmodell. Für diese Arbeit kommt es deshalb nicht weiter in Betracht.

2.1.6.2.5 Simulation und Test

Es werden Kriterien für die Überprüfung des Datenbankkonzepts aufgezeigt und Anleitungen für den Test gegeben.

2.1.6.3 Zusammenfassung

In dem Ansatz von Koreimann wird die logische Struktur der Daten sehr stark an den Erfordernissen eines hierarchischen Datenmodells orientiert.
Kritisch zu beurteilen ist dabei folgendes:
alle Datenelemente einer Anwendung werden durchnumeriert und zwar nach der Rangfolge, in der sie noch Ordnungskriterium sein können, d.h. es wird nicht nur unterschieden, ob ein Element Suchkriterium ist oder werden kann, sondern es wird ein u.U. unnötiger Aufwand getrieben, um eine Ordnungsrelation zwischen allen Attributen - auch den Nichtschlüsselattributen - und den jeweiligen Anwendungen zu finden.
In einigen Fällen wird eine so detaillierte Klassifizierung nur sehr schwer möglich sein. Eine solche Rangfolge aller Attribute wird kaum nennenswerte Vorteile für die Strukturierung der Daten bringen.

2.1.7 GEGENÜBERSTELLUNG VON RECHENERGEBNISSEN

2.1.7.1 Allgemeines

Um weitere Erkenntnisse über die Verwendbarkeit der Clusteranalyse zur Datenstrukturierung zu gewinnen, wurde aus der Vielzahl der Clusteranalyse-Verfahren das graphentheoretische Verfahren von Freudenthaler und Maier herangezogen, da es bisher die besten Ergebnisse geliefert hat und es den Erfordernissen des Relationenmodells am nächsten kommt.
Als Dateneingabe wurde zunächst eine überschaubare Matrix ausgewählt, an der die Ergebnisse demonstriert werden sollen. Dafür bot sich die Matrix A von Freudenthaler und Maier an.[29)]
Sodann wurden größere Datenverwendungsmatrizen berechnet und zum Vergleich der aufgeführten Verfahren herangezogen. Es handelt sich dabei um die Matrix B [30)], die ebenfalls im Verfahren von Freudenthaler und Maier vorgestellt wird, ferner um die Matrix K[31)] aus dem Beispiel von Koreimann und um die Matrix D [32)] aus dem Verfahren von Döringer.
Darüberhinaus wurden verschiedene Abstands- und Ähnlichkeitsmaße herangezogen um zu zeigen, daß diese bereits ganz erheblich das Ergebnis der Clusteranalyse beeinflussen können.

---

29) vgl. Freudenthaler, B., Maier, B.: Die Entwicklung eines Datenmodells, a.a.O., S.524
30) vgl. ebenda, S.527
31) vgl. Koreimann, D.S.: Anwendungsanalyse, a.a.O., S.3
32) vgl. Döringer, H.: Eine Methodik zur Strukturierung von Datenbanken, a.a.O., S.94

2.1.7.2 Ergebnisse am Beispiel der Matrix A

Besonderheiten der Matrix A:
Die Matrix A enthält 2 gleiche Zeilen (Zeile 3 und 5), so daß man von der Clusteranalyse erwarten darf, daß mindestens diese beiden Zeilen in einem Cluster zusammengefaßt werden. Ferner gibt es zwei Zeilen, die keine Übereinstimmung in den 1-Elementen aufweisen (Zeile 1 mit 3, oder 1 mit 5, oder 1 mit 4). Daher dürfen diese Zeilen nicht zusammengefaßt werden. Ähnliches gilt für die Spalten; Spalte 1 und 2 sind gleich, Spalte 5 und 1, oder 5 und 2, oder 5 und 3 haben keine Übereinstimmung in den 1-Elementen.

Das Verfahren von Freudenthaler und Maier liefert den Wert 2 als "bestmögliches" Klassifikationsniveau bei zeilen- und spaltenweiser Klassifikation (hier sind ausnahmsweise beide Werte gleich).
Auf diesem Klassifikationsniveau ergibt sich bei zeilenweiser Klassifikation (wie bereits im Kapitel 2.1.2 gezeigt wurde) eine mögliche Clusterbildung mit den Zeilen 3,1,5 und 4,2.
Da die Zeilen 3 und 5 gleich sind, müssen sie in jedem Fall zusammengefaßt werden. Zeile 4 und 2 sind ähnlich und gehören damit ebenfalls in ein Cluster.
Anders ist es bei der Zeile 1, die hier den Zeilen 3 und 5 zugeordnet ist, obwohl sie in keinem 1-Merkmal mit den anderen Zeilen übereinstimmt.
Insoweit ist das Ergebnis also fragwürdig.
Zum Vergleich wurde die Matrix A mit allen Varianten des Klassifikationsniveaus berechnet (vgl. Anhang 1 bis 10).
Das Klassifikationsniveau 3 liefert für die Zeile 1 ein besseres Ergebnis (vgl. Anhang 4), die Zeile 1 wird hier einem separaten Cluster zugeordnet.
Beim maximalen Klassifikationsniveau, dem Klassifikationsniveau 4 (vgl. Anhang 5), werden nur noch gleiche Zeilen zusammengefaßt.

Aus diesem Beispiel ist bereits erkennbar, wie sehr das Ergebnis der Clusteranalyse von dem gewählten Klassifikationsniveau und dadurch von der Anzahl der zu bildenden Gruppen abhängig ist.
Bei spaltenweiser Klassifikation der Matrix A liefert die "Baumbildung" kein eindeutiges Ergebnis (vgl. Anhang 7). Durch disjunkte Klassenbildung (sie wurde im Versuchsprogramm nicht implementiert, weil sie für das Ergebnis der Datenanalyse nicht ausschlaggebend ist) wäre ein denkbares Ergebnis

Cluster 1: 1,2,3
Cluster 2: 4

Bei näherer Betrachtung ist dieses Ergebnis nicht zufriedenstellend. Die Spalte 3 stimmt nur in einem 1-Element mit den Spalten 1 und 2 überein (vgl. Anhang 1). Es wäre daher besser folgende disjunkte Klassen zu bilden:

Cluster 1: 1,2
Cluster 2: 3
Cluster 3: 4

Um eine bessere Entscheidungshilfe zu finden, wurde auch die transponierte Matrix A mit den Klassifikationsniveaus von 1 bis 5 berechnet (vgl. Anhang 6 bis 10).
Dabei zeigt sich, daß auch hier das Klassifikationsniveau 3 ein besseres Ergebnis liefert.
Aus dem bisher gesagten geht hervor, daß Ergebnisse aus Clusteranalysen nicht ohne genaue Kenntnis des Datenmaterials verwendet werden sollten.

2.1.7.3 Ergebnisse am Beispiel der Matrix B

Als "bestmögliches" Klassifikationsniveau erhält man nach dem Verfahren von Freudenthaler und Maier (vgl. Anhang 11-24) bei zeilenweiser Klassifikation den Wert 13 und für die spaltenweise Klassifikation den Wert 22. Die Baumbildung auf diesen beiden Klassifikationsniveaus ist bereits so unübersichtlich (vgl. Anhang 15 und 24), daß sich diese Werte nicht mehr ohne weiteres in disjunkte Klassen einteilen lassen. Als Entscheidungshilfe mag die Baumbildung auf höheren Klassifikationsniveaus dienen. Deshalb sind die Baumbildungen für d transponierte Matrix auf den Klassifikationsniveaus 16 und 28 berechnet worden (vgl. Anhang 23 und 24). Aus dem Anhang 24 erkennt man, daß die Spalten 5, 12 und 14 auf jeden Fall zusammengefaßt werden sollten. Für die Matrix B wurden auch noch andere Abstands- und Ähnlichkeitsmaße berechnet um zu sehen, ob auch hier dasselbe, wie für die Matrix A gilt (vgl. Anhang 16 bis 20). Auch hier kann man erkennen, daß beim TANIMOTO-Abstand und beim S-Koeffizienten und seinen Varianten die Ähnlichkeiten bzw. die Abstände zu wenig differenziert werden. Die besten Ergebnisse liefert nach wie vor das bei Freudenthaler und Maier verwendete Ähnlichkeitsmaß und der Matching-Koeffizient (vgl. Anhang 12 und 17).

2.1.7.4 Ergebnisse am Beispiel der Matrix D

Für dieses Beispiel (vgl. Anhang 25 bis 32) wurden nur 0/1-Elemente verwendet. Um eine Einteilung in Schlüssel- und Nichtschlüsselattribute vorzunehmen, wie sie bei Döringer gewünscht wird, müßte man eine Gewichtung der Attribute einführen. Die Ergebnisse lassen sich jedoch bei nicht gewichteten Attributen besser vergleichen, deshalb wurde an dieser Stelle auf eine Gewichtung verzichtet.
Da diese Datenverwendungsmatrix sehr viele gleiche Zeilen und Spalten enthält, wurde ein Weg gesucht diese Zeilen und Spalten von vornherein zusammenzufassen. Auf dem höchstmöglichen Klassifikationsniveau enthält die Baumbildung immer gleiche Zeilen/Spalten.
Im Beispiel der Matrix D erkennt man beim Klassifikationsniveau 15 die gleichen Zeilen (vgl. Anhang 29):

Zeile 8,9,10 11,12
5,6
15,16
17,19
20,21,22

Bei den Spalten (vgl. Anhang 32) sind es die

Spalten 3,5
11,15

Die Baumbildung auf dem "bestmöglichen" Klassifikationsniveau 19(bei spaltenweiser Klassifikation) liefert Ergebnisse, die ähnlich dem Ergebnis bei Döringer sind: (vgl. Anhang 30):

Cluster 1: 1,3,5,13
Cluster 2: 6,2,4
Cluster 3: 8,12
Cluster 4: 10,14
Cluster 5: 11,15

Die Spalte 7 müsste entweder dem Cluster 1 zugeordnet werden, oder es müßte ein Cluster mit 1,3 und 7,5,13 gebildet werden.
Wenn man, wie bei Döringer nur drei Cluster bildet, führt das in jedem Fall dazu auch weniger ähnliche Elemente noch zusammenzufassen.

2.1.7.5 Ergebnisse am Beispiel der Matrix K

Auch dieses Beispiel (vgl. Anhang 33 bis 37) wurde nur mit 0/1-Elementen durchgerechnet. Auf dem nach Freudenthaler und Maier bestmöglichen Klassifikationsniveau (Wert 9 bei zeilenweiser Klassifikation, vgl. Anhang 35) erhält man folgende Gruppierung:

Cluster 1: 1,2,3
2: 5,4,14,15,16
3: 7,11,24
4: 23,6,9,8
5: 10,12,17,18
6: 13,19,20,22,21

Bei spaltenweiser Klassifikation erhält man auf dem Klassifikationsniveau 17 (vgl. Anhang 37):

Cluster 1: 1,4
2: 5,10,13
3: 7,8
4: 12,2,3,6,9,11

2.1.7.6 Zusammenfassung

Aus den vorausgehenden Beispielen ist ersichtlich, wie schwierig eine Klassifikation für größere Datenverwendungsmatrizen ist. Bereits bei einer Matrix mit mehr als 20 Zeilen oder Spalten, ist das Ergebnis nur schwer überprüfbar.
Es ist keineswegs sicher, maschinell die beste Lösung zu erhalten.
Ein möglicher - allerdings sehr rechenaufwendiger - Weg wäre die Gruppierung beginnend beim maximal möglichen Klassifikationsniveau vorzunehmen und diese Gruppe durch Reduzierung des Klassifikationsniveaus immer weiter zu vervollständigen. Das Klassifikationsniveau wäre dann solange zu reduzieren, bis alle Zeilen/Spalten der Matrix einem Cluster zugeordnet sind.
Da es sich bei der Strukturierung der Daten für ein Datenbanksystem im allgemeinen um umfangreiches Datenmaterial handelt - bei dem auch vorab keineswegs alle Anforderungen bekannt sind - scheint es wenig sinnvoll Relationen anhand der Clusteranalyse einer Datenverwendungsmatrix zu bilden. Der Rechenaufwand ist verhältnismäßig groß und die Ergebnisse hängen stark vom gewählten Ähnlichkeitsmaß ab.
Eine weitere Schwierigkeit ergibt sich daraus, daß in Relationen, die aufgrund einer Datenverwendungsmatrix gefunden werden, die Attribute in dieser Relation nicht zwingend funktional abhängig vom Schlüsselattribut sein müssen.

2.2 BESTIMMUNG VON RELATIONEN DURCH GENERALISIERUNG

2.2.1 ZIELSETZUNG

Das Datenmaterial für eine aufzubauende Datenbank ist im allgemeinen sehr komplex. Um nun den Datenbankadministrator nicht von vornherein mit Details zu konfrontieren, wird in diesem Ansatz zunächst von abstrakten Datengruppen ausgegangen. Diese werden nach bestimmten Entwurfskriterien schrittweise verfeinert. Auf diese Weise werden Relationen gefunden, die anschließend in einem Normalisierungsprozeß in 3NF überführt werden müssen. [33]

33) vgl. Smith,J.M., Smith,D.C.P.: Data Base Abstraction. In: Data Base Design Methodology, an Intensive Short Course. Engineering Summer Conf. Ann Arbor, Michigan 1976, S.25
und Chen, P.P., Yao, S.B.: Design and Performance Tools for Database Systems. In: Proc. of the 3rd Int. Conf. on Very Large Data Bases. New York 1977, S.6
und Smith, D.C.P., Smith, J.M.: Conceptual Data Base Design. In: INFOTECH State of the Art Report. Data Design, Invited Papers. Maidenhead 1980, S. 148 ff

2.2.2 VORGEHENSWEISE

2.2.2.1 Bildung disjunkter Objektklassen

Die Methode von Smith und Smith wird anhand von Beispielen - die diesem Artikel entnommen wurden - beschrieben.[34]
Zunächst wird eine generelle Klassifikation des Objekts vorgenommen. Diese wird bei Smith und Smith am Beispie einer Klassifikation von Fahrzeugen gezeigt:

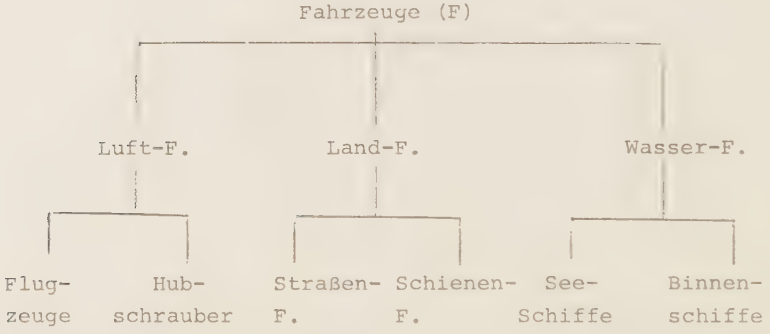


Bild 18: Hierarchische Klassifikation

Um aus einer solchen Klassifikation Relationen im Sinne von Codd zu bilden, müssen Merkmale der Objekt-Typen (Attribute) gefunden werden, die allen Objekt-Typen gemeinsam sind. Diese könnten für Fahrzeuge z.B. Identifikationsnummer, Hersteller, Preis und Nutzlast o.ä. sein. Zusätzlich können auch Attribute gebildet werden, die nicht allen, sondern nur bestimmten Objekt-Typen gemeinsam sind.
Im obigen Beispiel wurde die Typisierung so vorgenommen, daß dadurch disjunkte Klassen entstanden sind. Da diese Eigenschaft normalerweise nicht Vorraussetzung für eine Klassifikation ist, könnte eine andere Klassifikation ohne

---

34) vgl. Smith, J.M., Smith, D.C.P.: Database Abstractions: Aggregation and Generalization. ACM Transaction on Database Systems. 2 (1977), S.105-133

weiteres zu nicht disjunkten Klassen führen:

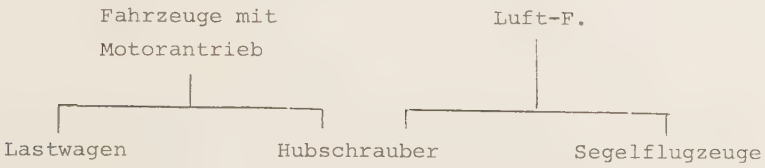


Bild 19: Klassifikation ohne disjunkte Klassen [35]

Die von Smith und Smith vorgeschlagene Methode fordert jedoch unbedingt disjunkte Klassen.
Auf diese Forderung kann nur dann verzichtet werden, wenn die Konsistenz der Daten gewährleistet, daß keine Fehler beim Änderungsdienst (speziell beim Einfügen oder Löschen) auftreten. [36]
Abhängig davon aus welcher und wessen Sicht die zu klassifizierenden Objekte betrachtet werden, können die unterschiedlichsten Typen gebildet werden. Je differenzierter die Typenbildung vorgenommen wird, desto eher ist gewährleistet, daß dadurch alle Eigenschaften der Objekte berücksichtigt werden. [37]

---

35) vgl. Smith, J.M., Smith, D.C.P.: Database Abstraction: Aggregation and Generalization, a.a.O., S.110
36) vgl. Housel, B.C., Waddle, V., Yao, S.B.: The Functional Dependency Model for Logical Database Design. In: Proc. 5th Int. Conf. on Very Large Databases. Rio de Janeiro 1979, S. 198
37) vgl. Weldon, J.-L.: Using Data Base Abstractions for Logical Design. The Computer Journal, 23, 1980, S. 43

2.2.2.2 Bildung von Relationen

Aus den Attributen und Namen der Objekt-Typen können auf jeder Stufe der Typenbildung Relationen gefunden werden. So können zunächst aus den allgemeinen Attributen von Fahrzeugen Relations-Typen gebildet werden. Ebenso können jedoch Relations-Typen aus den charakteristischen Attributen von beispielsweise Hubschraubern gebildet werden. Es ist ja leicht einsehbar, daß für Hubschrauber andere Merkmale von Bedeutung sind als etwa für Binnenschiffe oder Schienenfahrzeuge.
Smith und Smith bilden nun auf jeder Stufe der Klassifikation die typischen Merkmale (Attribute) und legen die charakteristischen Attribut-Werte fest. Auf diese Weise werden auf jeder Stufe der Klassifikation unterschiedlich detaillierte Attribute zu Relationen zusammengefaßt.

2.2.2.3 Bestimmung einer allgemeingültigen Struktur

Wird das Sammeln von Informationen und die Verallgemeinerung der Datenelemente getrennt erarbeitet, so können nur einfache Zusammenhänge erfaßt werden. Nur wenn beides im Zusammenhang gesehen wird, können alle Varianten eines Modells betrachtet werden.
Für den Objekt-Typ "Arbeitnehmer" seien die typischen Attribute festgelegt worden, so z.B. das Attribut "Beruf". Nun kann man für die Attributwerte dieses Attributs wiederum die typischen Merkmale bestimmen. So haben die Berufe "Sekretärin" oder "Chauffeur" unterschiedliche, charakteristische Merkmale.
Liegen nun alle Attribute der Berufsgruppen vor, dann könnnen ähnliche Attribute und ähnliche Berufsgruppen wieder zusammengefaßt werden.
Auf diese Weise werden Relationen gebildet, deren Attribute genau den Detaillierungsgrad haben, der erforderlich ist um alle später benötigten Informationen aus der Datenbank wiedergewinnen zu können.

2.2.3 ZUSAMMENFASSUNG

Die Vorgehensweise von Smith und Smith ermöglicht es dem Datenbankadministrator, mehr Semantik in den Entwurfsprozeß einzubeziehen, als dies bei den sonst in der Literatur zu findenden Methoden der Fall ist. [38] Auch das Entity-Relationship-Modell von Chen liefert als Ergebnis Relationen, die anschließend in einem Normalisierungsprozeß in 3NF transformiert werden müssen. Die Methode von Chen - auf die im dritten Kapitel noch kurz eingegangen wird - läßt dem Datenbankadministrator jedoch mehr individuelle Entscheidungsfreiheit, als dies bei der Methode von Smith und Smith der Fall ist.
Mit dieser - rechentechnisch nicht unterstützten - Methode können Relationen gefunden werden, deren Attribute funktional oder auch voll funktional abhängig voneinander sind. Wohingegen dies bei den im Kapitel 2.1 gezeigten Methoden nicht zwingend der Fall war. Dennoch sind rein manuelle Methoden bei komplexen Anwendungen zu unhandlich. [38]

---

38) vgl. Krieger, R., Lausen, G., Stucky,W.: Kurzfassung zum Vortrag "Methodischer Entwurf von Datenstrukturen für betriebliche Informationssysteme", o.O., o.J., S.3f
und Chen, P.P.: The Entity-Relationship-Model - Toward a Unified View of Data. ACM Transaction on Database Systems, 1 (1976), S.9-36
und Krieger, R., Lausen, G., Stucky,W.: Methodischer Datenentwurf beim Aufbau betrieblicher Informationssysteme. Angewandte Informatik 1982, S. 192 f

2.3 RELATIONEN-SYNTHESE

2.3.1 ALLGEMEINES

Ausgangspunkt bei der Relationen-Synthese ist ein Erhebungsprozeß, [39] bei dem zunächst alle voll funktionalen Abhängigkeiten der zu verarbeitenden Attribute beschrieben werden. Die so ermittelten Relationen werden Elementare Funktionalrelationen (EFR) genannt. Aus diesen Elementaren Funktionalrelationen (EFR) werden schrittweise immer größere Relationen zusammengesetzt, die letzlich den Anforderungen der optimalen 3NF genügen müssen. Das setzt voraus, daß ein Normalisierungsprozeß durchgeführt wird.
In den nachfolgend aufgeführten Beispielen werden die EFR in Form eines Graphen dargestellt. Diese Darstellungsform dient der besseren Anschaulichkeit (allerdings nur für eine geringe Anzahl von Attributen).
In einem 1. Schritt können alle EFR mit denselben linken Seiten zu einer Relation zusammengefaßt werden. So geht das in Kapitel 2.3.5 beschriebene Verfahren davon aus, daß dieser 1. Schritt bereits durchgeführt wurde und beginnt deshalb mit einer Adjazenzmatrix.
Die linken Seiten einer EFR können sowohl aus einzelnen Attributen als auch aus Attributgruppen bestehen. Im Kapitel 2.3.3 wird ein Verfahren gezeigt, das solche Attributgruppen auf der linken Seite einer EFR berücksichtigt, wohingegen bei dem in Kapitel 2.3.2 beschriebenen Verfahren vorausgesetzt wird, daß die Schlüssel (linken Seiten einer EFR) aus einzelnen Attributen bestehen.

---

39) vgl. Wedekind, H.: Datenbanksysteme I. Mannheim 1974, S.71

Allen nachfolgend beschriebenen Verfahren ist gemeinsam, daß sie nach diesem 1. Schritt - bei dem EFR mit denselben linken Seiten zu größeren Relationen zusammengefaßt werden - durch das Eliminieren überflüssiger Relationen einen Normalisierungsprozeß durchführen.
Ziel ist dabei Relationen in optimaler 3NF bzw. in der 4NF [40] zu finden.

---

40) vgl. Fagin, R.: Multivalued Dependencies and a New Normalform for Relational Databases. ACM Transaction on Database Systems, 8 (1977), S.262-278

2.3.2 SYNTHETISCHER ANSATZ NACH WANG UND WEDEKIND [41]

2.3.2.1 Zielsetzung

Durch Synthese Elementarer Funktionalrelationen (EFR) zu größeren Relationen soll eine minimale Überdeckung gefunden werden, welche allen Anforderungen zu genügen hat, die an die logische Struktur einer Datenbank gestellt werden.
Durch die Bestimmung voll funktionaler Abhängigkeiten wird eine Menge transitiv abgeschlossener EFR gefunden, die viele redundante - aus anderen EFR ableitbare - EFR enthält.
Ziel ist es nun, die redundanten EFR zu eliminieren um dadurch eine minimale Überdeckung zu erreichen. [42]
Diese minimale Überdeckung ist in dem Sinn optimal, daß es keine Anordnung mit weniger Relationen gibt. [43]

---

41) vgl. Wang, C.C., Wedekind, H.: Segment Synthesis in Logical Data Base Design. IBM Journal Research and Development, 19 (1975), S.71-77
und Wedekind, H.: Datenbanksysteme I, a.a.O., S.73ff

42) vgl. Delobel, C.: A Theory About Data in an Information System. IBM Research Report RJ964. San José Jan. 1972, S.26

43) vgl. Wedekind, H.: Datenbanksysteme I, a.a.O., S.82

2.3.2.2 Vorgehensweise

2.3.2.2.1 Festlegen der Elementaren Funktionalrelationen

Zunächst werden alle funktionalen Abhängigkeiten der Attribute erfaßt z.B. A → B. Es wird also festgelegt, welche Attribute A_i die Attribute B_i identifizieren. Die Attribute A_i sind Schlüsselattribute.
Eine EFR kann - wie bereits erwähnt - auf der linken Seite aus zusammengesetzten Attributen bestehen, während auf den rechten Seiten einzelne Attribute stehen, z.B. AB → C.
Im Beispiel von Wang und Wedekind werden nur einzelne Attribute als Schlüsselattribute verwendet. Zusammengesetzte Attribute - auf der linken Seite einer EFR - werden später in dem Beispiel von Bernstein behandelt.
Für das Beispiel bei Wedekind [44] mit den Attributen:

PNR = Angestellten Nr.
B = Berufsbezeichnung
ANR = Abteilungsnr. des Angestellten
MNR = Angestelltennr. des Abteilungsleiters
VT = Vertragstyp

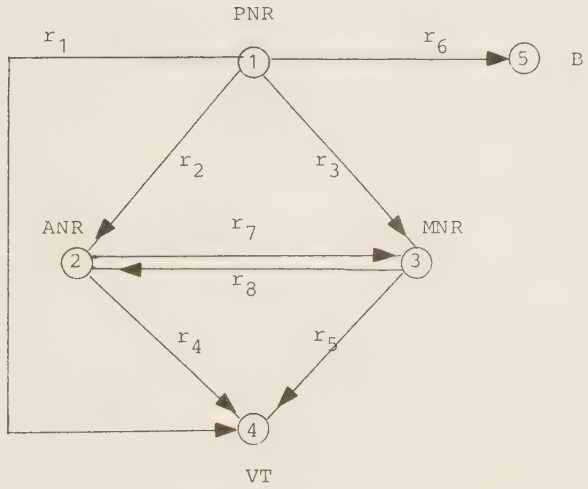
ergeben sich folgende Elementare Funktionalrelationen r_i:

---

44) vgl. Wedekind, H.: Datenbanksysteme I, a.a.O., S.75 und Codd, E.F.: Further Normalization of the Data Base Relational Model. Courant Computer Science Symposium "Data Base Systems", hrsg. von R.Rustin. New York 1971, S.47

r_1: PNR → VT
r_2: PNR → ANR
r_3: PNR → MNR
r_4: ANR → VT
r_5: MNR → VT
r_6: PNR → B
r_7: ANR → MNR
r_8: MNR → ANR

Aus diesen EFR läßt sich folgender Graph zeichnen:



2.3.2.2.2 Bestimmung der maximalen Distanz einer Relation

Unter der maximalen Distanz der Relation (PNR,VT) versteht man die maximale Anzahl Kanten, um von PNR nach VT zu gelangen, [45] also die Kanten r_3, r_8, r_7 und r_5 oder r_2, r_7, r_8 und r_4. Die Distanz für PNR → VT ist also 4 (bzw. unendlich, da r_7 und r_8 beliebig oft durchlaufen werden können [46]).
An dieser Stelle sei darauf hingewiesen, daß diese Distanzen nicht manuell vorgegeben werden müssen, sondern sich maschinell aus den angegebenen EFR ermitteln lassen. Ein Datenbankadministrator muß also lediglich die funktionalen Abhängigkeiten der Attribute kennen und vorgeben.

---

45) vgl. Wedekind, H.: Datenbanksysteme I, a.a.O., S.74f

46) Bei der maschinellen Bestimmung der maximalen Distanz ist das Ergebnis immer dann unendlich, wenn der Graph Zyklen enthält. Die maximale Distanz zwischen PNR und VT ist nur unter der Voraussetzung 4, daß der Zyklus ANR - MNR nur einmal durchlaufen wird.

2.3.2.2.3 Bestimmung der transitiven Zusammensetzung von Relationen

Die transitive Zusammensetzung $T(r_1,r_2) = r_3$ besagt, daß die Relationen r_1 und r_2 zusammengesetzt, die Relation r_3 ergibt.
Diese transitiven Zusammensetzungen können ebenso wie die Distanzen der Relationen aus den angegebenen EFR maschinell ermittelt werden.
Für das Beispiel von Wedekind erhält man [47)]

$T(r_i,r_j) = r_p$

$T(r_2,r_4) = T(r_3,r_5) = r_1$

$T(r_3,r_8) = r_2$

$T(r_2,r_7) = r_3$

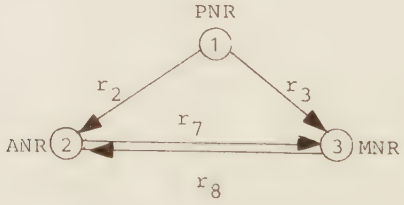
$T(r_7,r_5) = r_4$

$T(r_8,r_4) = r_5$

47) vgl. Wedekind, H.: Datenbanksysteme I, a.a.O., S.76 und Vetter, M., Maddison, R.N.: Database Design Methodology. Englewood Cliffs, New Jersey. 1981, S. 16 ff

2.3.2.2.4 Bestimmung der minimalen Überdeckung

Aus einer transitiv vollständigen Menge werden sukzessiv solange redundante Abhängigkeiten gestrichen, bis eine minimale Überdeckung erreicht ist. [48] Dabei können nur solche redundanten EFR gestrichen werden, die in einer minimalen Überdeckung mit Sicherheit nicht vorkommen dürfen. Beispielsweise muß eine transitive Zusammensetzung $T(r_2, r_3) = r_1$ bestehen, damit die Relation r_1 gestrichen werden kann.
Die Relation r_1 könnte jedoch nicht gestrichen werden, wenn sie Bestandteil einer anderen transitiven Zusammensetzung wäre (z.B. $T(r_1, r_9) = r_{10}$). Würde in einem solchen Fall r_1 gelöscht, so würde die transitive Zusammensetzung von r_{10} zerstört; r_{10} müßte dann erhalten bleiben, obwohl r_{10} über mehrere Relationen hinweg noch transitiv redundant sein könnte.
Um derartige Fehler zu vermeiden, beginnt man mit der Relation, die die maximale Distanz hat.
Lediglich bei einem tie kann eine Relation gelöscht werden, obwohl sie noch Bestandteil einer transitiven Zusammensetzung ist:



Bei einem tie gibt es mehrere minimale Überdeckungen. Durch das Entfernen eines Elements im tie wird jeweils eine minimale Überdeckung zerstört.

---

48) vgl. Delobel, C.: A Theory About Data in an Information System, a.a.O., S.26ff
und Wedekind, H.: Datenbanksysteme I, a.a.O., S.78ff

Beispiel:

x_{71}	S_1^7	d_p	$T(r_i,r_j)$	M
	r_2	2	$T(r_3,r_8)$	x_{61}
	r_3	2	$T(r_2,r_7)$	x_{62}
	r_4	2	$T(r_7,r_5)$	x_{63}
	r_5	2	$T(r_8,r_4)$	x_{64}
	r_6	1	—	
	r_7	1	—	
	r_8	1	—	

Nachdem die Relation r_1 entfernt wurde liegt ein dreifacher tie vor. Daher gibt es vier gleichwertige Lösungszweige:

x_{61}	S_2^6	d_p	$T(r_i,r_j)$	M
	r_3	2	-	
	r_4	2	$T(r_7,r_5)$	x_{51}
	r_5	2	$T(r_8,r_4)$	x_{52}
	r_6	1	-	
	r_7	1	-	
	r_8	1	-	

x_{62}	S_3^6	d_p	$T(r_i,r_j)$	M
	r_2	2	-	
	r_4	2	$T(r_7,r_5)$	x_{53}
	r_5	2	$T(r_8,r_4)$	x_{54}
	r_6	1	-	
	r_7	1	-	
	r_8	1	-	

x_{63}	S_4^6	d_p	$T(r_i,r_j)$	M
	r_2	2	$T(r_3,r_8)$	x_{55}
	r_3	2	$T(r_2,r_7)$	x_{56}
	r_5	2	-	
	r_6	1	-	
	r_7	1	-	
	r_8	1	-	

x_{64}	S_5^6	d_p	$T(r_i,r_j)$	M
	r_2	2	$T(r_3,r_8)$	x_{57}
	r_3	2	$T(r_2,r_7)$	x_{58}
	r_4	2	-	
	r_6	1	-	
	r_7	1	-	
	r_8	1	-	

Nach dem nächsten Eliminationsschritt erhält man für die vier Lösungszweige:

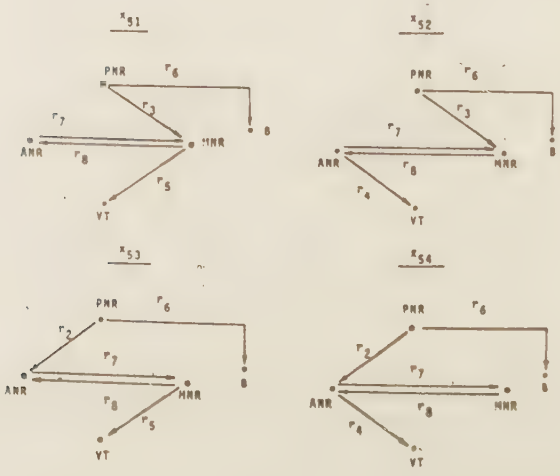
x_{51}	S_4^5	d_p	$T(r_i,r_j)$	M
	r_3	2	-	
	r_5	2	-	
	r_6	1	-	
	r_7	1	-	
	r_8	1	-	

x_{52}	S_5^5	dp	$T(r_i,r_j)$	M
	r_3	2	-	
	r_4	2	-	
	r_6	1	-	
	r_7	1	-	
	r_8	1	-	

x_{53}	S_4^5	d_p	$T(r_i,r_j)$	M
	r_2	2	-	
	r_5	2	-	
	r_6	1	-	
	r_7	1	-	
	r_8	1	-	

x_{54}	S_5^5	d_p	$T(r_i,r_j)$	M
	r_2	2	-	
	r_4	2	-	
	r_6	1	-	
	r_7	1	-	
	r_8	1	-	

Dabei bezeichnet x_{ij} die Tabellennummer, S_K die Menge der Relationen und d_p die Distanz der Relationen. Stellt man obige Tabellen in Form eines Graphen dar, so erhält man:



Die aufgezeigten Relationen befinden sich in der 3NF, da durch die minimale Überdeckung Transitivitäten ausgeschlossen wurden. Da es auch keine Anordnung mit weniger Relationen gibt, weil keine weiteren redundanten Abhängigkeiten eliminiert werden können, befinden sich die Relationen dieser vier Lösungen in der optimalen 3NF:

x_{51}
R1 (PNR,MNR,B)
R2 (MNR,ANR,VT)
R3 (ANR,MNR)

x_{52}
R1 (PNR,MNR,B)
R4 (ANR,MNR,VT)
R5 (MNR,ANR)

x_{53}
R6 (PNR,ANR,B)
R2 (MNR,ANR,VT)
R3 (ANR,MNR)

x_{54}
R6 (PNR,ANR,B)
R4 (ANR,MNR,VT)
R5 (MNR,ANR)

Diese vier Lösungen sind jedoch nur gleichwertig in Bezug auf die Anzahl der Relationen, nicht jedoch in Bezug auf die Anzahl der Zugriffe, die für das Auffinden eines Attributes erforderlich sind. Beispielsweise muß bei einem Zugriff auf VT über PNR bei den Lösungen x_{53} und x_{52} ein Zugriff (MNR-ANR) mehr erfolgen als bei den anderen Lösungen.

2.3.2.2.5 Überprüfung der Zugriffshäufigkeiten beim Änderungsdienst

In einer Relation der 3NF wird durch ein Tupel nur ein entity beschrieben, weshalb Änderungen, die sich auf ein entity beziehen, einfach durchzuführen sind. Neben dieser Forderung nach einfacher Änderungsmöglichkeit der entities besteht aber auch die Forderung, die Anzahl der Zugriffe auf ein Minimum zu beschränken.
Die Anzahl der Zugriffe, die für die Änderung eines entities erforderlich sind, kann unter Umständen in Relationen der 2NF geringer sein, als bei Relationen in der 3NF.[49] Dennoch hat die Forderung nach Relationen in 3NF höhere Priorität, da bei Relationen in der 1NF oder 2NF beim Änderungsdienst Fehler auftreten können, die als update-, insertion- oder deletion-dependency bekannt sind.
Das Verfahren von Wedekind berücksichtigt keine Zugriffshäufigkeit. Es wird jedoch im Anschluß an die Beschreibung der minimalen Überdeckung gezeigt, daß die Anzahl der erforderlichen Zugriffe bei Relationen in 2NF geringer sein kann, als bei Relationen in 3NF.

---

49) vgl. Wedekind, H.: Datenbanksysteme I, a.a.O., S.86ff

2.3.2.3 Zusammenfassung

Ausgehend von voll funktionalen Abhängigkeiten je zweier Attribute wird eine Beschreibung des Datenmaterials vorgenommen, die weitgehend von späteren Anwendungen unabhängig ist.
Durch diese voll funktionalen Abhängigkeiten ist das Datenmaterial eindeutig beschrieben. Insofern ist dieser Ansatz für den Datenbankadministrator außerordentlich günstig.
Aus den angegebenen funktionalen Abhängigkeiten werden Relationen in optimaler 3NF erzeugt. Die programmtechnische Lösung beim Auftreten mehrerer tie's ist jedoch sehr speicher- und rechenaufwendig, da alle Lösungsmöglichkeiten ermittelt werden.
Die Bestimmung der maximalen Distanz zwischen den Attributen, die Voraussetzung für die Ermittlung der minimalen Überdeckung ist, wird dann problematisch, wenn der Graph aus voll funktionalen Abhängigkeiten Zyklen enthält. Verwendet man in solchen Fällen allgemeingültige Algorithmen zur Bestimmung der Distanzen, so erhält man - wie bereits erwähnt - als maximale Distanz den Wert unendlich. Distanzen mit dem Wert unendlich sind aber für die Bestimmung der minimalen Überdeckung ungeeignet.

2.3.3 SYNTHETISCHER ANSATZ NACH Bernstein [50]

2.3.3.1 Zielsetzung

Ebenso wie bei dem Ansatz nach Wang und Wedekind werden aus vorgegebenen funktionalen Abhängigkeiten Relationen in optimaler 3NF entwickelt.
Dabei sollen folgende Anforderungen erfüllt werden:

1. die Relationen müssen vollständig sein, d.h. alle Attribute müssen in den Relationen vorhanden sein
2. keine Relation darf transitive Abhängigkeiten beinhalten
3. die Anzahl der Relationen soll minimiert werden.

Die bei Bernstein angegebenen Beispiele beschäftigen sich vorwiegend mit solchen EFR, die auf der linken Seite aus Attributgruppen bestehen. Dabei können im Normalisierungprozeß Komplikationen entstehen, die bei dem bisherigen Beispiel bewußt ausgeklammert wurden.
Bernstein fordert außerdem, daß das Ergebnis nicht nur den Anforderungen der 3NF sondern der BCNF genügt.
Beispiel:
für die EFR AB → C
und C → B
erhält man eine Relation in 3NF

R (A B C)

diese Relation ist nicht in der Boyce-Codd-Normalform (BCNF), da B voll funktional abhängig von C ist, nicht jedoch von A. [51]

---

50) vgl. Bernstein, P.A.: Synthesizing Third Normal Form Relations from Functional Dependencies. ACM Transactions on Database Systems, 1 (1976), S.277-298

51) vgl. Bernstein, P.A.: Normalization and Functional Dependencies in the Relational Data Base Model. Tech. Report CSRG-60, Computer Systems Res. Group. Department of Computer Science of the University of Toronto, Canada. 1975, S.18

2.3.3.2 Vorgehensweise

2.3.3.2.1 Algorithmus zur Ermittlung nichtredundanter Relationen

Die einfachste Lösung wäre, vorgegebene funktionale Abhängigkeiten so zu gruppieren, daß alle Attribute, die zu demselben Schlüsselkandidaten gehören, zusammengefaßt werden.
Dieses Vorgehen hat folgende Nachteile:

1. die so gefundenen Relationen sind nicht notwendigerweise in 3NF, da Transitivitäten nicht ausgeschlossen werden
2. die linken Seiten der EFR sind nicht zwingend Schlüsselattribute der gefundenen Relationen, wie in dem nachfolgenden Beispiel an der Relation R_4 gezeigt werden wird
3. die Anzahl der Relationen kann auf diese Weise nicht minimiert werden.

Der Algorithmus von Bernstein soll diese Nachteile verhindern.
Beispiel:[52)]
Gegeben seien die folgenden EFR:

f_1: A → B f_4: B → D
f_2: A → C f_5: D → B
f_3: B → C f_6: ABE → F

EFR mit denselben linken Seiten werden nun zu Gruppen zusammengefaßt:

$g_1 = (f_1, f_2)$ $g_3 = (f_5)$
$g_2 = (f_3, f_4)$ $g_4 = (f_6)$

Aus diesen Gruppen werden nun Relationen gebildet, die alle Attribute enthalten, die in diesen Gruppen vorkommen:

R1 (A,B,C) R3 (D,B)
R2 (B,C,D) R4 (A,E,B,F)

---

52) vgl. Bernstein, P.A., Beeri, C.: An Algorithmic Approach to Normalization of Relational Database Schemas. Technical Report CSRG-73. Computer Systems Research Group, University of Toronto, Canada. 1976, S.12
und Bernstein, P.A.: Synthesizing Third Normal Form Relations, a.a.O., S.283

Der Normalisierungsvorgang kann nun dadurch vereinfacht werden, daß zunächst redundante Attribute eliminiert werden.
Um eine nichtredundante Überdeckung zu finden, ist es keineswegs erforderlich, zunächst schwierigere EFR wie z.B. f_6 zu umgehen. Das Problem kann dadurch gelöst werden, daß überflüssige Attribute auf der linken Seite der EFR entfernt werden. [53)]
In obigem Beispiel ist f_2 redundant und kann entfernt werden, ohne daß die Bedingung der 3NF in R1 verletzt würde. Sind alle überflüssigen Elemente entfernt, so ist eine nichtredundante Überdeckung der EFR gefunden.
Das zweite Problem sind Relationen, deren Schlüssel funktional voneinander abhängig sind (inverse Abbildungen). Solche Relationen können in eine Relation überführt werden. [54)] Im Beispiel ist f_4 und f_5 invers. Die Relation R3 ist daher überflüssig und kann entfernt werden. Durch diese Vorbereitungen ist es möglich, die Relationen so zu bilden, daß dabei die Nachteile der o.a. "einfachen" Lösung vermieden werden.
Außerdem ist in f_6 B als Schlüsselkandidat infolge f_1 überflüssig.

---

53) vgl. Bernstein, P.A.: Synthezizing Third Normal Form, a.a.O., S.284
54) vgl. Bernstein, P.A.: Comment on "Segment Synthesis in Logical Data Base Design". IBM Journal of Research and Development, 20 (1976), S.412

2.3.3.2.2 Behandlung nichtfunktionaler Abhängigkeiten

Für nichtfunktionale Abhängigkeiten z.B. Auto und Fahrer (ein Fahrer kann unterschiedliche Autos fahren und ein Auto kann von unterschiedlichen Fahrern gefahren werden) wird ein Hilfsattribut Θ eingeführt. Wichtig ist dabei, daß zwar mehrere solcher Hilfsattribute in einer Menge von EFR eingeführt werden können, daß aber ein Hilfsattribut eindeutig einer Relation zugeordnet werden muß und nur in dieser Relation vorkommen darf.
Jede nichtfunktionale Abhängigkeit hat demnach ihr eigenes Θ - Element.
Beispiel: Gegeben seien:

f_1: AB $\rightarrow \Theta_1$

f_2: AB $\rightarrow \Theta_2$

In dem Algorithmus von Bernstein werden diese beiden funktionalen Abhängigkeiten zu einer Gruppe zusammengefaßt und daraus die Relation

$R\ (A,B,\Theta_1,\Theta_2)$

gebildet. Sofern jedoch nur eine nichtfunktionale Zuordnung für das Wertepaar A - B besteht, kann das Hilfsattribut Θ entfallen. [55)]

---

55) vgl. Bernstein, P.A.: Synthesizing Third Normal Form Relations, a.a.O., S.288
und Zaniolo, C., Melkanoff, M.A.: On the Design of Relational Database Schemata. ACM Transactions on Database Systems, 6 (1981), S. 39 f

2.3.3.2.3 Ermittlung der optimalen 3NF

Auch der Algorithmus von Bernstein liefert Relationen in optimaler 3NF (bzw. in BCNF), da transitive Zusammensetzungen eliminiert werden. In bestimmten Fällen liefert dieser Algorithmus jedoch weniger Relationen als bei Wang und Wedekind. Und zwar immer dann, wenn zwei Attribute voll funktional abhängig voneinander sind (inverse Abbildungen) würde der Algorithmus von Wang und Wedekind zwei Relationen erzeugen und der Algorithmus von Bernstein nur eine. Bernstein geht davon aus, daß eine dieser beiden Relationen überflüssig ist, weil sie aus der anderen abgeleitet werden kann. [56]

56) vgl. Bernstein, P.A.: Synthesizing Third Normal Form Relations, a.a.O., S.289
und Bernstein, P.A.: Comment on "Segment-Synthesis in Logical Data Base Design", a.a.O., S.412

2.3.3.3 Zusammenfassung

Bernstein behandelt bei seinen Beispielen vorwiegend solche funktionalen Abhängigkeiten, deren Schlüsselkandidaten aus Attributgruppen bestehen, wobei es vorkommen kann, daß diese Attributgruppen überflüssige Attribute enthalten.
Darüberhinaus werden mit diesem Algorithmus auch nichtfunktionale Abhängigkeiten behandelt. Diese werden durch Einführung eindeutiger Hilfsattribute in funktionale Abhängigkeiten überführt.
Im Gegensatz zu dem Verfahren nach Wang und Wedekind wird bei inversen Abbildungen nur eine Relation erzeugt.

2.3.4 SYNTHETISCHER ANSATZ NACH FAGIN [57]

2.3.4.1 Zielsetzung

Hier sollen nicht nur voll funktionale Abhängigkeiten in den Normalisierungsprozeß einbezogen werden, sondern auch "multivalued dependencies".
Außerdem wird - im Gegensatz zu den bisherigen synthetischen Ansätzen - die vierte Normalform (4NF) [58] angestrebt, die über die Anforderungen der 3NF und der BCNF hinausgeht.

---

57) vgl. Fagin, R.: Multivalued Dependencies and a New Normal Form for Relational Databases. ACM Transaction on Database Systems, 2 (1977), S. 262-278
und Fagin, R.: The Decomposition Versus the Synthetic Approach to Relational Database Design. In: Proc. of the 3rd Int. Conf. on Very Large Databases, Tokyo 3.-8. Okt. 1977. New York 1977, S.441-446

58) vgl. Fagin, R.: Multivalued Dependencies, a.a.O., S.267
" A relation schema R+ is in fourth normal form (4NF) if, whenever a nontrivial multivalued dependency X $\twoheadrightarrow$ Y holds for R+, then so does the functional dependency X $\rightarrow$ A for every column name A of R+. Intuitively all dependencies are the result of keys. In particular a 4NF relation schema can have no nontrivial multivalued dependencies that are not functional dependencies."

2.3.4.2 Vorgehensweise

2.3.4.2.1 Zum Begriff "multivalued dependencies"

Dieser Begriff ist für nachfolgendes Verfahren von grundsätzlicher Bedeutung. Bernstein's Verfahren geht von eindeutigen funktionalen Abhängigkeiten aus, so kann dort z.B. in der Relation

R (Arbeitnehmer, Kind)

ein Arbeitnehmer nur ein Kind haben, oder aber das Attribut Kind müßte in die Attribute 1. Kind, 2. Kind, 3. Kind usw. zerlegt werden. Bei Fagin würde die Abhängigkeit der Attribute Arbeitnehmer, Kind als "multivalued" behandelt und eine Zerlegung des Attributs Kind in 1.Kind, 2.Kind usw. wäre nicht erforderlich. Diese "multivalued dependencies" werden so in den Normalisierungsprozeß einbezogen, daß als Ergebnis Relationen in 4NF erreicht werden.

2.3.4.2.2 Normalisierung der "multivalued dependencies"

Aus den trivialen multivalued dependencies

AB ↠ C

erhält man die Relation [59]

R (A,B,C)

Wird nun eine Relation R+ gefunden, die nur solche Attribute enthält, die voll funktional vom Schlüsselattribut abhängig sind, so ist diese Relation in 4NF. Ein Relation in 4NF muß nicht notwendigerweise so weit wie irgend möglich zerlegt werden.
Beispielsweise sei die Relation [59]

R+ (A,B,C,D)

aus den EFR A ↠ B
A ↠ C
A ↠ D

gebildet. Die Attribute B,C und D sind demnach alle nur von dem Schlüsselattribut A abhängig.
Dann ist R+ in der 4NF, obwohl es möglich ist, R+ in

R1 (A,B)
R2 (A,C)
R3 (A,D)

ohne Verlust an Information zu zerlegen.

---

59) vgl. Fagin, R.: Multivalued Dependencies, a.a.O., S.267f

2.3.4.2.3 Beispiel für eine Normalisierung in 4NF [60]

Gegeben sei eine Relation, die alle vorkommenden Attribute enthält:

R (Schulklasse,Tag,Fach,Raum,Lehrbuch) (1.0)

diese Attribute haben folgende funktionale Abhängigkeit:

Schulklasse,Fach,Tag → Raum (1.1)

und multivalued dependencies:

Schulklasse, Fach →→ Lehrbuch (1.2)

Schulklasse, Fach →→ Tag, Raum (1.3)

Nimmt man an, daß in einer Schulklasse alle Schüler dieselben Bücher benützen, so kann man die Relation (1.2) auch als

Schulklasse → Lehrbuch (1.2.1)

schreiben.

Damit könne die multivalued dependencies zerlegt werden in:

R1 (Schulklasse, Lehrbuch)

R2 (Schulklasse, Fach, Tag, Raum)

Da nun alle Attribute voll funktional abhängig von den Schlüsselattributen sind, sind diese Relationen in 4NF.

Die Kritik van de Riets richtet sich vornehmlich gegen die in den Beispielen gewählte Festlegung der multivalued dependencies, die er als m:n Verbindungen definiert.

---

60) vgl. Fagin, R.: Multivalued Dependencies, a.a.O., S.269f
und Riet van de, R.P.: Normalization - A Tool in Designing a Database.In: INFOTECH State of the Art Report. Data Design, Invited Papers. Maidenhead 1980, S. 274 ff

2.3.4.3 Zusammenfassung

Der Vorteil dieses Verfahrens besteht darin, daß nicht nur eindeutige funktionale Abhängigkeiten sondern auch "multivalued dependencies" verarbeitet werden können. Fagin geht dabei davon aus, daß das Optimum der 4NF je nach Anwendungsfall verschieden sein kann. Durch die Definition der funktionalen und mehrwertigen Abhängigkeiten kann dieses Optimum beeinflußt werden. Fagin hält es daher für wünschenswert, daß der Anwender durch diese Definition selbst steuernd in den Normalisierungsprozeß eingreifen kann und dadurch, die für ihn günstigste Lösung erreichen kann.
Das Verfahren von van de Riet geht von multivalued independencies aus kommt jedoch zu äquivalenten Ergebnissen.

2.3.5 SYNTHETISCHER ANSATZ NACH VETTER [61]

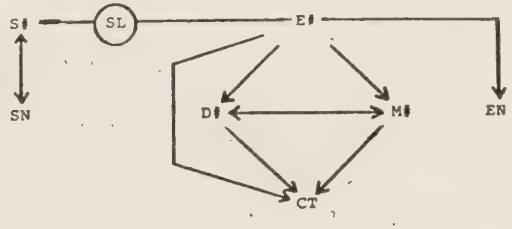
2.3.5.1 Zielsetzung

Dieses Verfahren setzt sich zum Ziel, die physische Datenstruktur aus einer geeigneten Beschreibung der Realität zu finden. Dabei soll die Anzahl der Zugriffe, die für eine Operation erforderlich ist minimiert werden.
Grundlage hierfür ist die Normalisierung der Ausgangs-Relationen in 3NF. Für dieses Kapitel soll das Verfahren nur in Bezug auf die Synthese der logischen Datenstruktur betrachtet werden.

2.3.5.2 Vorgehensweise

2.3.5.2.1 Festlegung der Ausgangs-Relationen

Das Verfahren soll an folgendem Beispiel erläutert werden:[62]



CT contract type
D# department number
E# employee number
EN employee name
M# manager number
S# skill number
SL skill level
SN skill name

---

61) vgl. Vetter, M.: Data Base Design by Applied Data Synthesis. Proc. of the 3rd. Int. Conf. on Very Large Data Bases. New York 1977, S.428-440
62) vgl. ebenda, S.431

Es wird eine Adjazenzmatrix M erstellt, die in den Zeilen und Spalten alle auftretenden Attribute enthält.
Alle Attribute, die von einem Schlüsselattribut aus erreicht werden können, werden in der Zeile der Matrix M mit 1 gekennzeichnet.

M	CT	D#	E#	E# * S#	EN	M#	S#	SL	SN
CT									
D#	1					1			
E#	1	1			1	1			
E#*S#			1				1	1	
EN									
M#	1	1							
S#									1
SL									
SN							1		

Bild 20: Matrix M

2.3.5.2.2 Behandlung von m:n Verbindungen

Nicht voll funktionale Abhängigkeiten, die bei Bernstein durch Einführung eines Hilfsattributs Θ aufgelöst werden, werden hier ebenfalls durch Einführung eines Hilfsknotens aufgelöst (vgl. dazu auch S. 124f).
Beispiel:

E# — S#

wird in

E# → D# x S#

und

E# x S# → S#

aufgelöst.
Dadurch sind die Relationen, die in den Zeilen der Matrix M dargestellt wurden bereits in der 2NF,da alle Attribute einer Relation voll funktional abhängig vom Schlüsselattribut sind.

2.3.5.2.3 Bestimmung der direkten und indirekten Abhängigkeiten

Um die Relationen in 3NF transformieren zu können, müssen transitive Abhängigkeiten eliminiert werden. Dazu ist zunächst festzustellen, für welche Nicht-Primärattribute mehrere Zugriffswege existieren. Deshalb wird untersucht, welche Nicht-Primärattribute indirekt von Primärattributen abhängen. Diese Abhängigkeiten werden in der Matrix R dargestellt:

$$R \leftarrow M \text{ v. } \wedge M$$

Die Adjazenzmatrix M wird also quadriert und r_{ij} erhält immer dann den Wert 1, wenn das innere Produkt einen Wert größer oder gleich 1 liefert.

R	CT	D#	E#	E# * S#	EN	M#	S#	SL	SN
CT									
D#	1	1							
E#	1	1				1			
E#*S#	1	1			1	1			1
EN									
M#	1					1			
S#							1		
SL									
SN									1

Bild 21: Matrix R

R ist somit eine Erreichbarkeitsmatrix und besagt, daß alle mit 1 gekennzeichneten Attribute über eine Distanz 2 erreicht werden können.
Da nun die direkten (Distanz=1) und die indirekten (Distanz=2) Abhängigkeiten gesucht werden, wird die Matrix T gebildet, die nur solche Elemente enthält, die in R und in M vorkommen:

$$T \leftarrow M \wedge R$$

T	CT	D#	E#	E# * S#	EN	M#	S#	SL	SN
CT									
D#	1								
E#	1	1				1			
E#*S#									
EN									
M#	1								
S#									
SL									
SN									

Bild 22:Matrix T

Dieser Weg, die indirekten Verbindungen von Attributen - über das innere Produkt der Adjazenzmatrix M - zu ermitteln, gilt jedoch nur für den Fall, daß eine direkte Verbindung zweier Attribute A-C durch eine indirekte Verbindung der Attribute A-B-C mit der Distanz 2 ersetzt werden kann:

Dies ist in dem angegebenen Beispiel immer der Fall. Würde eine direkte Verbindung zweier Attribute nur durch eine indirekte Verbindung mit der Distanz 3 ersetzt werden können, so müßte R nochmals mit M multipliziert werden.

2.3.3.2.4 Eliminieren von transitiven Abhängigkeiten

Um bei direkt und indirekt voneinander abhängigen Attributen feststellen zu können, welche Verbindung aufgelöst werden kann, ist es wichtig, zunächst einmal die Bedeutung der Relationen zu kennen.
Alle indirekten und direkten Verbindungen, die in der Matrix T durch den Wert 1 gekennzeichnet sind,

werden in der Matrix W gewichtet, dabei wurde für dieses Beispiel vereinbart, daß die Verbindung von umso größerer Bedeutung ist, je größer der Wert in der Matrix W ist.

W	CT	D#	E#	E# * S#	EN	M#	S#	SL	SN
CT									
D#	5								
E#	1	4				2			
E# * S#									
EN									
M#	3								
S#									
SL									
SN									

Bild 23: Matrix W

Nun wird in der Matrix M zunächst die Verbindung E#, CT gelöscht, da diese in W mit dem niedrigsten Wert gewichtet wurde. Dann wird aus der neuen Matrix M wiederum die Matrix R und T ermittelt. Danach wird in M das Element gelöscht, das in W den nächst höheren Wert hat und dieses Verfahren solange wiederholt, bis alle Elemente der Matrix T den Wert 0 haben; also solange, bis alle transitiven Verbindungen eliminiert sind. Dadurch befinden sich die Relationen, die in den Zeilen der neuen Matrix M abgebildet sind, in der 3NF.

2.3.3.3.1 Bestimmung der Optimalen 3NF

Es wird ein Optimum in dem Sinn gesucht, daß es keine Lösung mit weniger Relationen in 3NF gibt.
Die Matrix M liefert nach der ersten beschriebenen Variante zwar eine überflüssige Relation, [1] wenn zwei Relationen voll funktional abhängig voneinander sind. Sind zwei Attribute voll funktional abhängig voneinander, so muß zunächst entschieden werden, welches Attribut Primärattribut werden soll.
Die Zeile der Matrix M des Nicht-Primärattributs wird nun in die Zeile des Primärattributs kopiert, das heißt alle '1'-Elemente werden übernommen und die Zeile des Nicht-Primärattributs auf 0 gesetzt.
Beispiel:

M	CT	D#	S#	S# * S#	SN	N#	S#	SL	SN
CT									
D#	1	1				1			
S#		1			1		1		
S# * S#			1				1	1	
SN									
N#									
S#							1		1
SL									
SN									

---

[1] vgl. Bernstein, P.A.: Comment on "Recent Synthesis Algorithms for Relational Data Base Design". a.a.O., S. 1

[illegible]

In dem Verfahren von Yetter soll eine physische Datenbankstruktur gefunden werden. Aus diesem Grund wird überprüft ob die vom Yetter gefundenen Relationen allen bereits benannten Zugriffsanforderungen genügen.
Da für die vorliegende Arbeit der bei Yetter vorgeschlagene Normalisierungsprozeß von Interesse ist, wird hier nicht weiter auf die Überprüfung der Zugriffsanforderungen eingegangen.

[illegible]

Das Verfahren von Yetter ist durch die Matrixdarstellung der Funktionalen Abhängigkeiten zwar speicheraufwendig, dafür aber rechnerintern sehr einfach zu lösen.
Werden die indirekten Verbindungen der Attribute durch quadrieren der Adjazenzmatrix M gefunden, so gilt die Lösung nur für solche Beispiele, bei denen eine direkte Verbindung durch eine indirekte Verbindung , die immer die Distanz 2 hat, ersetzt werden kann. (was bei Yetter vorausgesetzt wird).
Dies ist bei allen Beispielen im Kapitel 2.3 der Fall, gilt jedoch nicht für alle Datenstrukturen.
Die Überprüfung der Zugriffsanforderungen entspricht der Überprüfung der Erreichbarkeit der Attribute auf der Basis bereits benannter Anforderungen.

2.3.6 GEGENÜBERSTELLUNG DER RECHENERGEBNISSE

2.3.6.1 Rechenergebnisse des ersten Beispiels

2.3.6.1.1 Ermittlung der 3NF nach dem Verfahren von Wang und Wedekind

Als erstes Beispiel wurde das Beispiel von Wedekind [64] gewählt. Dieses Beispiel wurde im Kapitel 2.3.2 bereits ausführlich beschrieben. Aus den dort aufgeführten funktionalen Abhängigkeiten erhält man als Adjazenzmatrix:

i \ j	PNR 1	ANR 2	MNR 3	VT 4	B 5
1	0	1	1	1	1
2	0	0	1	1	0
3	0	1	0	1	0
4	0	0	0	0	0
5	0	0	0	0	0

Dabei wurden die Datenelemente der Einfachheit halber durchnumeriert:

1 = PNR = Angestelltennr.
2 = ANR = Abteilungsnummer
3 = MNR = Angestelltennummer des Abteilungsleiters
4 = VT = Vertragstyp
5 = B = Berufsbezeichnung

Die maximalen Distanzen zwischen den Datenelementen sind untenstehender Matrix zu entnehmen:

i \ j	1	2	3	4	5
1	0	2	2	4	1
2	0	0	1	2	0
3	0	1	0	2	0
4	0	0	0	0	0
5	0	0	0	0	0

64) vgl. Wedekind, H.: Datenbanksysteme I, a.a.O., S.75

So ist die maximale Distanz zwischen PNR und VT in der Zeile 1 und Spalte 4 zu finden. [65)]
In der Reihenfolge der Distanzen - beginnend bei der maximalen Distanz - wurden nun die Relationen durchnumeriert:

i \ j	1	2	3	4	5
1	0	2	3	1	6
2	0	0	7	4	0
3	0	8	0	5	0
4	0	0	0	0	0
5	0	0	0	0	0

Die Relation r_1 setzt sich somit aus den Attributen PNR und VT (Zeile 1, Spalte 4) zusammen.
Äquivalente Relationen wurden so dargestellt, daß sie in der Zeile der Relation jeweils durch eine gleiche Ziffer in der entsprechenden Spalte gekennzeichnet sind:

i \ j	1	2	3	4	5	6	7	8
1	1	2	3	2	3	0	0	0
2	0	1	2	0	0	0	0	2
3	0	2	1	0	0	0	2	0
4	0	0	0	1	2	0	2	0
5	0	0	0	2	1	0	0	2
6	0	0	0	0	0	1	0	0
7	0	0	0	0	0	0	1	0
8	0	0	0	0	0	0	0	1

Bild 24: Matrix äquivalenter Relationen

65) hier wurde von den bei Wedekind angegebenen Distanzen ausgegangen. Es wird dabei berücksichtigt, daß der Zyklus ANR-MNR genau einmal durchlaufen wird, ansonsten wäre die maximale Distanz zwischen PNR und VT unendlich. Da in den hier betrachteten Beispielen ein Zyklus aus maximal 2 oder 3 Kanten besteht, konnte im Programm berücksichtigt werden, daß der Zyklus nur einmal durchlaufen wird.

So ist beispielsweise die Relation r_1 (Zeile 1) durch die Relationen r_2 und r_4 (Ziffer 2 in Spalte 2 und 4) oder durch die Relationen r_3 und r_5 (Ziffer 3 in Spalte 3 und Spalte 5) ersetzbar.
Das Ergebnis des Normalisierungsprozesses ist der untenstehenden Tabelle der äquivalenten Relationen zu entnehmen:

	1	2	3	4	5	6	7	8
1	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0
3	2	2		0	0	0	0	0
4	2	0	0	0	0	0	0	0
5	2	0	0	0	1	0	0	0
6	2	0	0	0	0		0	0
7	2	0	0	0	0	0		0
8	2	0	0	0	0	0	0	

Bild 25: reduzierte Matrix äquivalenter Relationen

Da der programmtechnische Aufwand unverhältnismäßig hoch wäre, um alle Lösungszweige zu ermitteln (es wären alle Kombinationen mit verhältnismäßig hohem Rechenzeit- und Speicherplatz-Aufwand zu ermitteln), wurde diese Matrix so belassen, daß alle Lösungsmöglichkeiten daraus abgelesen werden können.
Als erste Lösungsmöglichkeit bieten sich alle durch 1 gekennzeichneten Relationen an:

$r_1 r_4 r_6 r_8$

das entspricht der Lösung x_{51} bei Wedekind. [66]
Als zweite Lösungsmöglichkeit bietet sich an, die Relationen r_3 und r_5 durch die äquivalenten Relationen zu ersetzen.
Als Lösung erhält man dann:

$r_2 r_3 r_5 r_7 r_8$

das entspricht der Lösung x_{54} bei Wedekind.
Die beiden anderen Lösungen entstehen daraus, daß wahlweise nur r_3 bzw. nur r_5 durch die äquivalenten Relationen ersetzt wird:

$r_2 r_3 r_6 r_7 r_8$ bzw. $r_2 r_5 r_6 r_7 r_8$

was den Lösungen x_{53} und x_{52} bei Wedekind entspricht.

---

66) vgl. Wedekind, H.: Datenbanksysteme I, a.a.O., S.79 und Kapitel 2.3.2.2.4

Im Anhang 38 bis 45 sind die Ergebnisse für das erste Beispiel mit unterschiedlichster Gewichtung aufgeführt. An der Vielzahl der verwendeten Gewichtungen soll demonstriert werden, daß es nicht ohne weiteres möglich ist, mit beliebig gewählter Gewichtung alle Lösungszweige, die bei dem Verfahren von Wedekind gezeigt wurden, zu ermitteln. Da für dieses Beispiel die gewünschten Lösungen bekannt waren, wurde die Gewichtung so gewählt, daß diese Lösungen erzielt wurden. Ist die gewünschte Lösung nicht bekannt, so ist es u.U. möglich eine Lösung zu finden, die nicht in optimaler 3NF ist (vgl. Anhang 45). Andererseits führt eine Vielzahl der Gewichtungen - wie im Anhang 38 bis 44 dargestellt - zu derselben Lösung.
Für inverse Abhängigkeiten werden bei Wedekind - wie bereits erwähnt - zwei Relationen erzeugt, bei dem Verfahren von Vetter werden in solchen Fällen - entsprechend der Gewichtung - die funktionalen Abhängigkeiten des Nicht-Primärattributs in die Relation des Primärattributs übernommen. Dadurch wird nur eine Relation erzeugt (vgl. Anhang 38 bis 44).

2.3.6.2 Rechenergebnisse des zweiten Beispiels

2.3.6.2.1 Ermittlung der Anordnung nach dem Verfahren von Hoffer und Severance

Als zweites Beispiel wurde das - beim Verfahren von Vetter verwendete - gewählt.[88]
Die Adjazenzmatrix ergibt sich wiederum aus den funktionalen Abhängigkeiten:

	A#	B#	C#	D#/E#	F#	G#	H#	I#	J#
.	1	2	3	4	5	6	7	8	9
1	0	0	0	0	0	0	0	0	0
2	1	0	0	0	0	1	0	0	0
3	1	1	0	0	1	1	0	0	0
4	0	0	0	0	0	0	1	1	0
5	0	0	0	0	0	0	0	0	0
6	1	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	1	0	0

Bild 28: Adjazenzmatrix für das zweite Beispiel

Auch hier wurden die Datenelemente der Einfachheit halber durchnumeriert.
Als maximale Distanzen zwischen den Attributen erhält man (vgl. 2.3.6.1):

88) Vgl. Vetter, M.: Data Base Design by Applied Data Synthesis, a.a.O., S.431
und Kapitel 2.3.5

i \ j	1	2	3	4	5	6	7	8	9
1	0	0	0	0	0	0	0	0	0
2	2	0	0	0	0	1	0	0	0
3	4	2	0	0	1	2	0	0	0
4	0	0	1	0	0	0	1	1	0
5	0	0	0	0	0	0	0	0	0
6	2	1	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	1
8	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	1	0	0

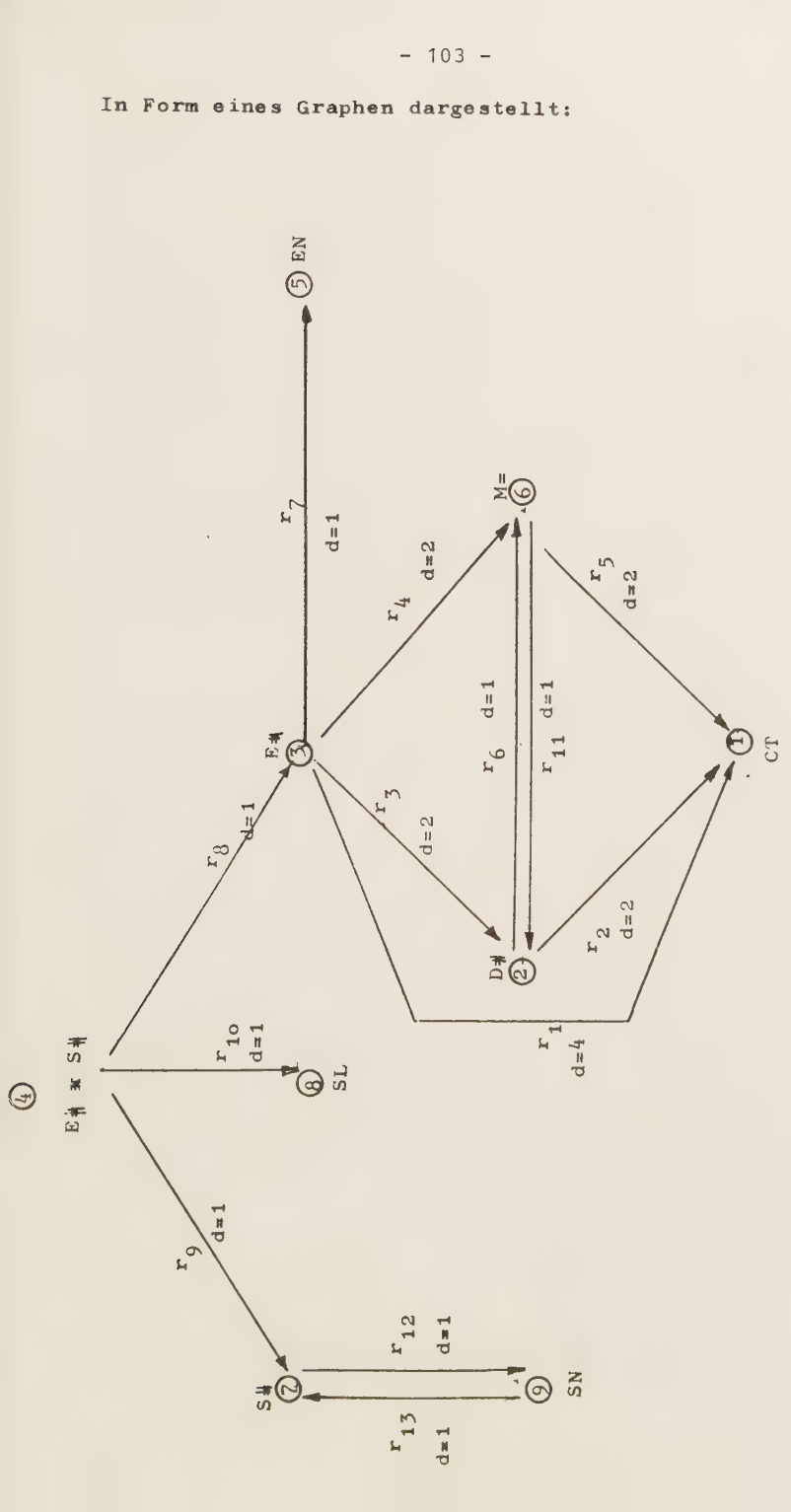
Bild 29: maximale Distanzen für das zweite Beispiel

Die Relationen wurden in der Matrix R in der umgekehrten Reihenfolge der maximalen Distanzen durchnumeriert:

i \ j	1	2	3	4	5	6	7	8	9
1	0	0	0	0	0	0	0	0	0
2	2	0	0	0	0	6	0	0	0
3	1	3	0	0	7	4	0	0	0
4	0	0	8	0	0	0	9	10	0
5	0	0	0	0	0	0	0	0	0
6	5	11	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	12
8	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	13	0	0

Bild 30: Matrix R für das zweite Beispiel

In Form eines Graphen dargestellt:



d = maximale Distanz
r_x = Nr. der Relation
(n) = Knotennummer (Matrixzeile bzw. Spalte in M)

Daraus ergeben sich folgende äquivalente Relationen:

i \ j	1	2	3	4	5	6	7	8	9	10	11	12	13
1	1	2	2	3	3	0	0	0	0	0	0	0	0
2	0	1	0	0	2	2	0	0	0	0	0	0	0
3	0	0	1	2	0	0	0	0	0	0	2	0	0
4	0	0	2	1	0	2	0	0	0	0	0	0	0
5	0	2	0	0	1	0	0	0	0	0	2	0	0
6	0	0	0	0	0	1	0	0	0	0	0	0	0
7	0	0	0	0	0	0	1	0	0	0	0	0	0
8	0	0	0	0	0	0	0	1	0	0	0	0	0
9	0	0	0	0	0	0	0	0	1	0	0	0	0
10	0	0	0	0	0	0	0	0	0	1	0	0	0
11	0	0	0	0	0	0	0	0	0	0	1	0	0
12	0	0	0	0	0	0	0	0	0	0	0	1	0
13	0	0	0	0	0	0	0	0	0	0	0	0	1

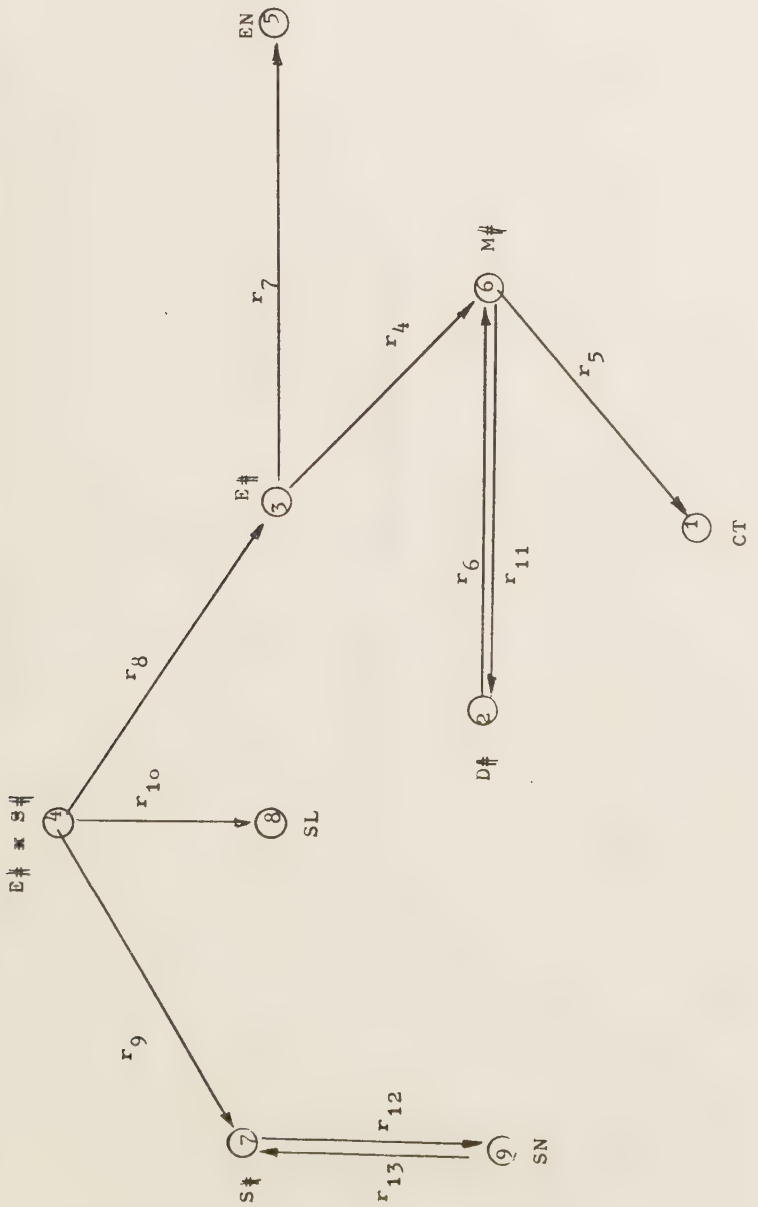
die möglichen Lösungen sind nachfolgender Matrix der reduzierten äquivalenten Relationen zu entnehmen:

i \ j	1	2	3	4	5	6	7	8	9	10	11	12	13
1	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	2	1	0	2	0	0	0	0	0	0	0
5	0	2	0	0	1	0	0	0	0	0	2	0	0
6	0	0	0	0	0	1	0	0	0	0	0	0	0
7	0	0	0	0	0	0	1	0	0	0	0	0	0
8	0	0	0	0	0	0	0	1	0	0	0	0	0
9	0	0	0	0	0	0	0	0	1	0	0	0	0
10	0	0	0	0	0	0	0	0	0	1	0	0	0
11	0	0	0	0	0	0	0	0	0	0	1	0	0
12	0	0	0	0	0	0	0	0	0	0	0	1	0
13	0	0	0	0	0	0	0	0	0	0	0	0	1

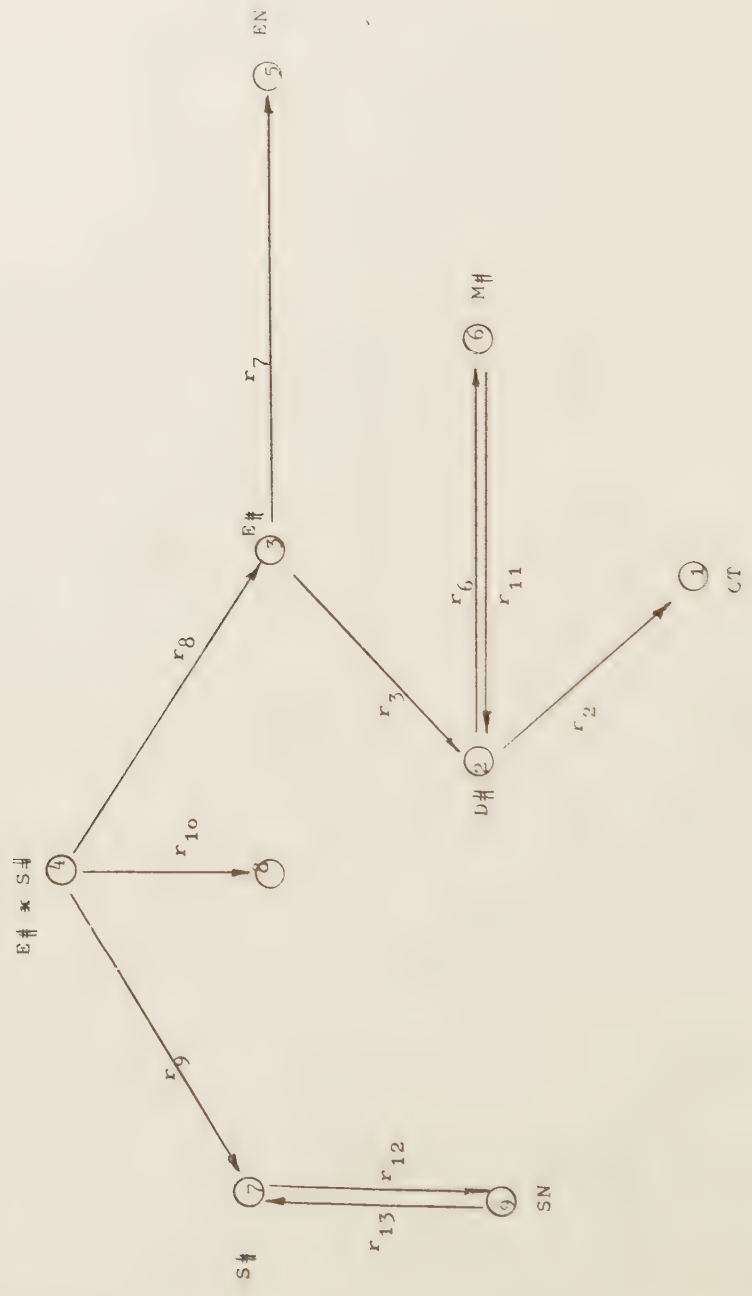
Wie bei dem ersten Beispiel in Kapitel 2.3.6.1 erhält man durch die Zeile 4 und 5 vier Lösungsmöglichkeiten, die nachfolgend graphisch dargestellt wurden.

Inverse Abhängigkeiten liefern hier jeweils zwei Relationen (r_{12},r_{13} und r_6,r_{11}).

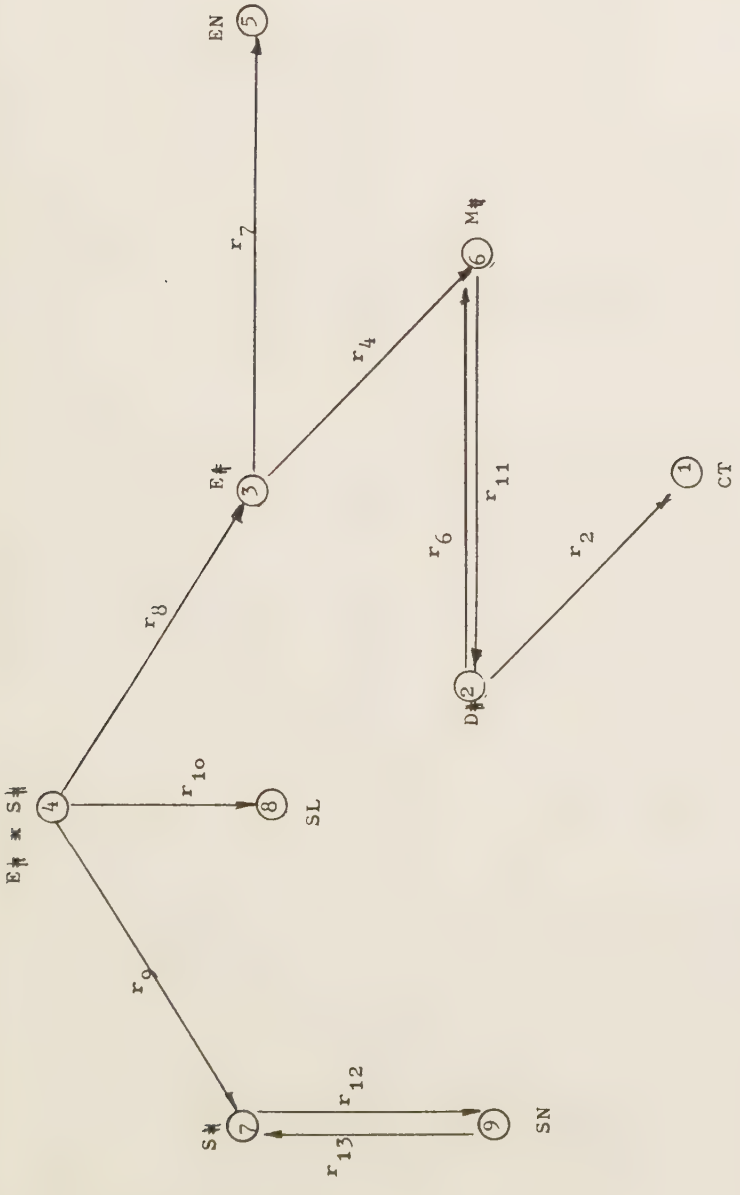
1. Lösungsmöglichkeit:



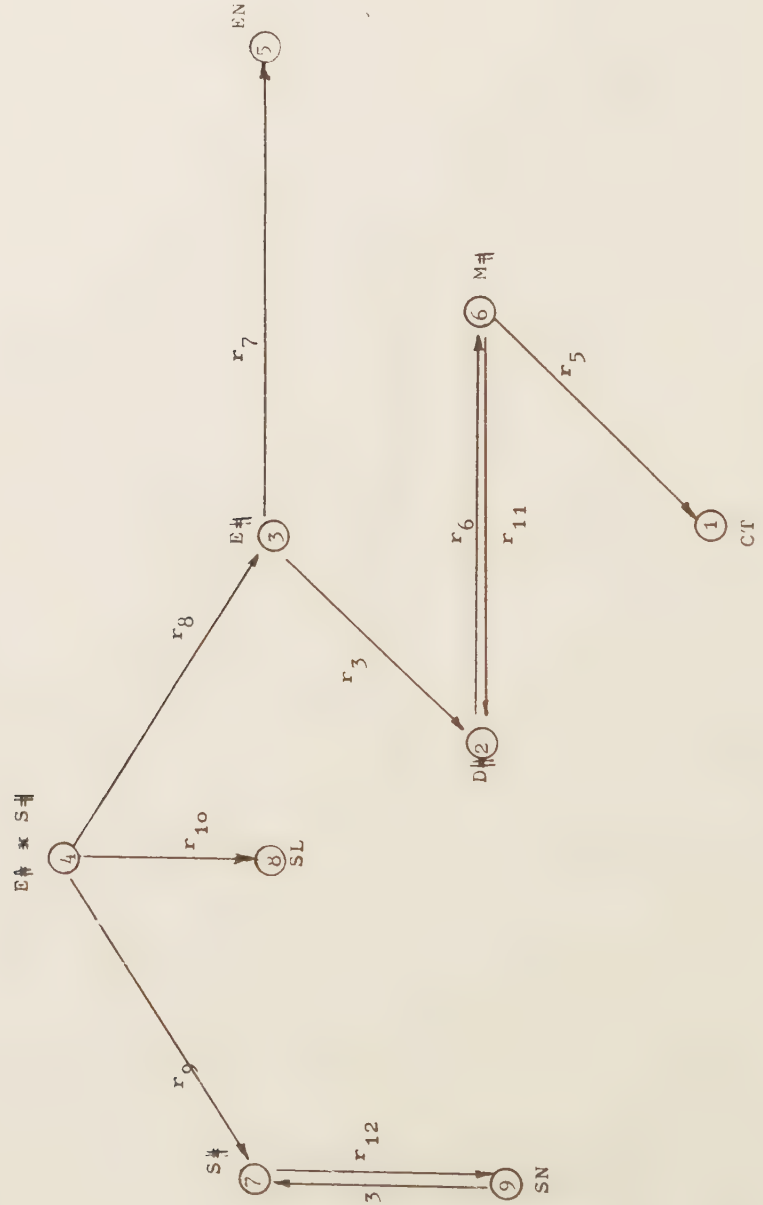
2. Lösungsmöglichkeit



3. Lösungsmöglichkeit



4. Lösungsmöglichkeit:



2.3.6.2.2 Ermittlung der 3NF nach dem Verfahren von Vetter

In der Matrix R wurden die indirekten Abhängigkeiten für das zweite Beispiel dargestellt:

i \ j	1	2	3	4	5	6	7	8	9
1	0	0	0	0	0	0	0	0	0
2	1	1	0	0	0	0	0	0	0
3	2	1	0	0	0	1	0	0	0
4	1	1	0	0	1	1	0	0	1
5	0	0	0	0	0	0	0	0	0
6	1	0	0	0	0	1	0	0	0
7	0	0	0	0	0	0	1	0	0
8	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	1

und in der Matrix T die Attribute, die indirekt und direkt voneinander abhängig sind:

i \ j	1	2	3	4	5	6	7	8	9
1	0	0	0	0	0	0	0	0	0
2	1	0	0	0	0	0	0	0	0
3	1	1	0	0	0	1	0	0	0
4	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0
6	1	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0

Als Gewichtung W wurde verwendet:

i \ j	1	2	3	4	5	6	7	8	9
1	0	0	0	0	0	0	0	0	0
2	5	0	0	0	0	0	0	0	0
3	1	4	0	0	0	2	0	0	0
4	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0
6	3	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0

Bei dem Verfahren von Vetter werden inverse Abbildungen zu jeweils einer Relation zusammengefaßt, man erhält dann - mit obiger Gewichtung - als Lösung:

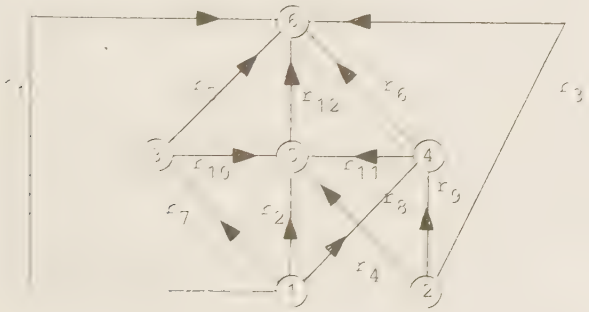
i \ j	1	2	3	4	5	6	7	8	9
1	0	0	0	0	0	0	0	0	0
2	1	1	0	0	0	1	0	0	0
3	0	1	0	0	1	0	0	0	0
4	0	0	1	0	0	0	1	1	0
5	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	1	0	0
8	0	0	0	0	0	0	0	0	1
9	0	0	0	0	0	0	0	1	1

Im Anhang 46 bis 50 wurden die Ergebnisse, die sich aufgrund willkürlich gewählter Gewichtungen ergaben, zusammengestellt. Das Ergebnis der noch nicht zusammengefaßten Relationen im Anhang 46 entspricht der 4.Lösungsmöglichkeit, wie sie auf S. 108 dargestellt wurde. Die Gewichtung im Anhang 47 und 49 liefert ein Ergebnis, das der 2. Lösungsmöglichkeit entspricht. Die Gewichtungen im Anhang 48 und 50 weisen - für die bereits zusammengefaßten Relationen - zumindest mangelnde Erreichbarkeit der Knoten auf. Betrachtet man jedoch die Ergebnisse im Anhang 48 und 50, bei denen inverse Abhängigkeiten noch nicht zusammengefaßt wurden, so stellt man fest, daß daß diese Ergebnisse nicht in optimaler 3NF sind, da sie redundante Abhängigkeiten enthalten.
Im Anhang 48 gibt es keine Verbindung mehr vom Knoten 3 nach 2, stattdessen gibt es eine direkte Verbindung vom Knoten 3 nach 1. Wäre 3-2 vorhanden, so wäre der Knoten 1 von Knoten 3 aus erreichbar, bei dieser Lösung ist Knoten 2 von 3 aus nicht erreichbar.Entsprechendes gilt für das Ergebnis im Anha 50. Dort fehlt die Verbindung von Knoten 6 nach 1, dafür gibt es eine Verbindung von Knoten 3 nach 1. Dadurch geht die Erreichbarkeit von Knoten 6 nach 1 verloren.

2.3.6.3 Rechenergebnisse des dritten Beispiels [69]

2.3.6.3.1 Ermittlung der BNF nach dem Verfahren von Wang und Wedekind

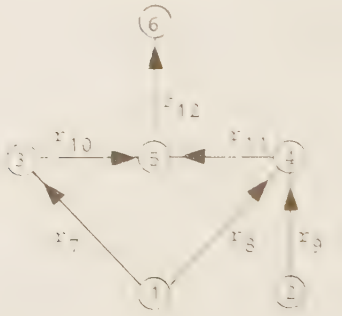
Als drittes Beispiel wurde gewählt:



Charakteristisch für dieses Beispiel ist, daß es zwei Wurzeln (Knoten 1 und 2) hat.

69) vgl. Dörfler, W.: Mathematik für Informatiker, Band 1. München 1977, S.63

Eine der möglichen Lösungen ist dann (vgl. Anhang 51):



An diesem Beispiel ist ersichtlich, daß es nicht mehr einfach ist, die möglichen Lösungen aus der Matrix der äquivalenten Relationen abzulesen.

2.3.6.3.2 Ermittlung der 3NF nach dem Verfahren von Vetter

Die gewählte Gewichtung und die Lösung ist im Anhang 52 dargestellt. Die Lösung ist identisch mit der oben dargestellten Lösung nach Wang und Wedekind.
Diese Lösung ist jedoch nicht in 3NF, da sie redundante EFR in den Relationen r_7, r_{10} und r_8,r_{11} aufweist. Wird jedoch die Relation r_{10} entfernt, so ist der Knoten 6 oder 5 von 3 aus nicht mehr erreichbar.

2.3.6.4 Zusammenfassung

Aus den vorangestellten Beispielen ist erkennbar, daß das Hauptproblem bei dem Verfahren von Vetter in der zu wählenden Gewichtung liegt. Es wurden auch Beispiele berechnet, bei denen die gewählte Gewichtung (vgl. Anhang 45, 48, 50 und 52) nicht zu zufriedenstellenden Ergebnissen geführt hat. Das Verfahren von Vetter ist so ausgelegt, daß eine Kante eines Graphen immer genau durch zwei andere Kanten ersetzt werden soll - sofern dies möglich ist. Dies dürfte die Ursache für die nicht zufriedenstellende Lösung des dritten Beispiels sein. Bei dem Verfahren von Wang und Wedekind ist die Ermittlung der verschiedenen Lösungsmöglichkeiten mit hohem Speicherplatz- und Rechenzeit-Aufwand verbunden. Der Fall der redundanten EFR r_7, r_{10} und r_8, r_{11} im dritten Beispiel ist beim Verfahren von Wang und Wedekind nicht explizit beschrieben.

3. STRUKTURIERUNG DES VORHANDENEN DATENMATERIALS FÜR EIN RELATIONALES DATENBANKSYSTEM

3.1 ZIELSETZUNG

Im zweiten Kapitel wurden einige Methoden zur Strukturierung des Datenmaterials untersucht. Im dritten Kapitel soll eine neue Methode entwickelt werden, die die Strukturierung des Datenmaterials für eine relationale Datenbank auf möglichst einfache Art rechentechnisch unterstützt. Es wird vorausgesetzt, daß das Verfahren Relationen liefert, die mindestens den Anforderungen der optimalen 3NF genügen. Um den Aufwand im Änderungsdienst der Datenbank so gering wie möglich zu halten, soll jedes Attribut nur durch genau eine funtkionale Abhängigkeit bestimmt werden, so daß neben den transitiven Abhängigkeiten auch Zyklen und inverse Abhängigkeiten eliminiert werden. Die Erreichbarkeit der Attribute darf dadurch jedoch nicht verlorengehen.Obwohl durch den Normalisierungsprozeß die Abhängigkeit der Relationen von der Zugriffshäufigkeit verringert wird [70] soll untersucht werden, ob und inwieweit es sinnvoll ist die Häufigkeit des Datenzugriffs im Normalisierungsprozeß zu berücksichtigen.
Das Verfahren soll möglichst rechenzeitgünstig sein, [71] daher wird zunächst als Ergebnis eine der zulässigen Lösungen ermittelt. Prinzipiell können mit diesem Verfahren jedoch alle zulässigen Lösungen ermittelt werden.
Damit kein Informationsverlust entsteht, sollen alle eventuell in Frage kommenden Schlüsselkandidaten als Schlüssel erhalten werden. Das heißt, daß die Anzahl der gesuchten Relationen der Anzahl der möglichen Schlüsselkandidaten entspricht.
Es wird die Einschränkung vorgenommen, daß Schlüsselkandidaten nur aus einfachen - nicht aus zusammengesetzten - Attributen bestehen und der Graph aus voll funktionalen Abhängigkeiten nicht mehr als eine Wurzel hat.

---

70) vgl. Codd, E.F.: Further Normalization, a.a.O., S. 34
71) vgl. Bernstein, P.A.: Normalization and Functional Dependencies in the Relational Data Base Model, a.a.O., S.45ff
und Härder, Th.: Zur Synthese von Relationen in optimaler dritter Normalform. Forschungsbericht DVI 76-3 Darmstadt 1976, S.4 und S.8ff

Um zu zeigen, wo das rechentechnische Verfahren in den Gesamtablauf des logischen Datenbankdesign einzuordnen ist, soll zunächst die allgemeine Vorgehensweise beim Entwurf logischer Datenstrukturen grob umrissen werden.
Die elementaren Ziele des Datenbank-Design [72] wie

- Datenunabhängigkeit
- Zugriffspfadunabhängigkeit und
- Redundanzminimierung

sowie

- Vermeidung von Inkonsistenzen
- Ausgleich widerspüchlicher Anforderungen
- Datensicherheit

sind bei der Konzeption selbstverständlich zu beachten.

---

72) vgl. Mertens, P., Griese, J.: Datenbanken und Datenbanksysteme. In: Handwörterbuch der Betriebswirtschaftslehre Bd. 1/1, hrsg. von E. Grochla und W. Wittmann. Stuttgart 1974, Spalte 1089
und Schönenberger, P., Soldan, C.P.: Datenunabhängigkeit im Anwendungsbereich von Datenbanken. Output 1981, Heft 6, S. 30 ff
und Wedekind, H.: Datenbanksysteme I, a.a.O., S. 54
und Österle, H.: Zum Entity-Relationship-Ansatz im Requirement Engineering. Angewandte Informatik 1982, S. 413

3.2 SYSTEMANALYSE

3.2.1 ALLGEMEINE VORGEHENSWEISE BEIM DATENBANK-DESIGN

Die logische Strukturierung der zu entwickelnden Datenbank läßt sich - ebenso, wie die allgemeingültige Gliederung einer Systemstudie - in die Phasen: [73]

1. Allgemeine Orientierung über das Problem
2. Wahl passender Entscheidungskriterien
3. Formulierung der Entscheidungsalternativen
4. Auswahl relevanter Umweltfaktoren
5. Modellkonstruktion und Modellvalidierung
6. Schätzung und Prognose der Modellparameter
7. Deduktion von Informationen aus dem Modell

einteilen.
Ähnliche Gliederungen, die mehr auf das spezielle Problem zugeschnitten sind, findet man z.B. bei Yao: [74]

1. Requirements Analysis
2. View Modelling
3. View Integration
4. View Restructuring
5. Schema Analysis and Mapping

oder bei Kahn: [75]

1. Requirements Step
2. Entity Step
3. Relationship Step
4. Entity Structure Step
5. Refinement Step
6. DBMS Step

um nur einige zu nennen. Die allgemeingültige Gliederung nach Hanssmann, wird in dieser Arbeit für die weitere Vorgehensweise gewählt, weil sie in einigen Punkten (z.B. Wahl passender Entscheidungskriterien) zu intensiverer Analyse zwingt, als die spezielleren Gliederungen nach Yao oder Kahn.

---

73) vgl. Hanssmann,F.: Einführung in die Systemforschung. München 1978, S.25f
74) vgl. Yao. S.B. et al.: An Integrated Approach to Logical Database Design. In: NYU Symposium on Data Base Design. New York 1978, S.1
75) vgl. Kahn, B.K.: A Structure Logical Database Design Methodology. In: NYU Symposium on Database Design. New York 1978, S.19

3.2.2 ORIENTIERUNG ÜBER DAS PROBLEM

Diese Orientierungsphase ist ein Teil der Anforderungsanalyse bei Yao oder Kahn. Hier sind zunächst die Gegebenheiten der realen Welt zu analysieren.
Um die Gesamtzusammenhänge verstehen und aufzeigen zu können sind folgende Fragen von Interesse:

1. Welche Bereiche sollen in der Datenbank abgebildet werden?
2. Welches Datenbankmodell kann dafür eingesetzt werden?
3. Welche Informationen werden zum Aufbau der Datenbank benötigt?
4. Welche Informationen stehen zur Verfügung?
5. Welche Probleme sollen durch die Datenbank gelöst werden (Zielsetzung)?
6. Welche Restriktionen gibt es?

Die Frage, welche Bereiche in der Datenbank abgebildet werden sollen, ist von dem jeweiligen Datenbankprojekt abhängig. Diese Frage hat jedoch letztlich keinen Einfluß darauf, wie die Datenbank logisch strukturiert werden soll; daher wird hier nicht näher darauf eingegangen.
Es bleibt lediglich anzumerken, daß eine Datenbank, die zuviele Bereiche umfaßt, schwer zu pflegen ist, daher kann es günstiger sein, eine Datenbank pro Bereich aufzubauen und - soweit möglich - die getrennten Datenbanken bei Suchfragen durch die Operation join zu verbinden. Dennoch gilt auch hier der Grundsatz des Datenbank-Aufbaus, die Daten so zu zentralisieren, daß möglichst viele Verarbeitungsprozesse auf diese zugreifen können. [76)]
Die zweite Frage, welches Datenbankmodell eingesetzt werden soll, bezieht sich darauf welche Datenbanksoftware und welche Hardware zur Verfügung steht.

---

76) vgl. Wedekind, H.: Relationale Datenbanksysteme. Informatik-Spektrum, 1 (1978), S.5

Da die relationalen Datenmodelle immer mehr an Bedeutung gewinnen, nicht zuletzt deshalb, weil weitgehend problemlos neue Attribute aufgenommen werden können, wird im Rahmen dieser Arbeit nur auf das Relationenmodell eingegangen.
Die dritte Frage, welche Informationen benötigt werden, kann man durch die Analyse der Informationsbedürfnisse mit Hilfe folgender Fragestellungen beantworten: [77)]

- welche Anforderungen an die Informationen sind bekannt
- welche Anforderungen werden vermutlich später noch gestellt
- welche Anforderungen sind sinnvoll
- wie können diese Anforderungen klassifiziert werden
- in welchen zeitlichen Abständen werden Informationen benötigt
- welche Informationen werden als Eingabe erforderlich

Welche Informationen zur Verfügung stehen, kann durch die

- Analyse der Arbeitsabläufe
- Analyse des Informationsflusses
- Analyse der integrativen Beziehungen

ermittelt werden.
Die Frage nach den Problemen, die durch die Datenbank gelöst werden sollen, hängt von der Zielsetzung der Datenbank-Anwendung ab. Neben den allgemeinen Grundsätzen eines Datenbankaufbaus wie Datenunabhängigkeit, Redundanzminimierung, Datenintegrität, Datensicherheit und -Schutz müssen die speziellen Ziele der jeweiligen Datenbankanwendung in organisatorischer und wirtschaftlicher Hinsicht analysiert werden. Eine Hilfestellung bei der

---

77) vgl. Grochla, E., Garbe, H., Gillner, R.: Gestaltungskriterien für den Aufbau von Datenbanken. Opladen 1973, S.11
und Bubenko jr., J.A.: Data Models and their Semantics. In: INFOTECH State of the Art Report. Data Design, Invited Papers. Maidenhead 1980, S. 130
und Vetter, M.: Systematische Entwicklung von Informationssystemen. Output 1981, Heft 6, S.39

Definition der speziellen Zielsetzung bieten folgende Fragen:

- welche der angestrebten Informationsgewinnung ist vorprogrammierbar, welche wird nur spontan möglich sein
- welche Engpässe gibt es bei der bisherigen Informationsgewinnung
- welche Informationen können bisher noch gar nicht geliefert werden

Die speziellen Ziele der Datenbankanwendung sollten sehr detailliert formuliert werden, da sich hieran auch der Nutzen dieser neu zu schaffenden Datenbank ableiten läßt. Bestehende Restriktionen können sowohl aus der Zielsetzung abgeleitet werden, als auch durch die Ermittlung der Struktur und die Bestimmung der besonderen Eigenschaften der abzubildenden Bereiche. Die allgemeinen Probleme einer Sollkonzeption wie z.B. psychologische Barrieren, Kommunikationsschwierigkeiten u.ä. sind selbstverständlich zu beachten. [78)]

Als Ergebnis der vorangegangenen Punkte muß feststehen, welche Benutzer welche Anforderungen an die zu organisierende Datenbank stellen. Im Allgemeinen wird man dabei Überschneidungen feststellen können und es wird die Aufgabe des Datenbankadministrators sein, diese Anforderungen so zu strukturieren, daß Überschneidungen minimiert werden.

Voraussetzung dafür ist die Bereitschaft, alle Anforderungen in Frage zu stellen, nicht zuletzt um daraus Erkenntnisse für eine sinnvolle Koordination zu finden. [79)] Wird in übertriebenem Maß auf die Benutzerfreundlichkeit des Kommunikationsprozesses geachtet, so besteht die Gefahr, daß die Wirtschaftlichkeitsgesichtspunkte der Datenbankanwendung vernachlässigt werden. [80)]

---

78) vgl. Bubenko, J.A., et al: From Information Requirements to DBTG-Data Structures. In: Proc. of the ACM SIGMOD-SIGPLAN Conf. on Data, Abstraction, Definition and Structure, hrsg. von H. Ledgard und R. Taylor. Salt Lake City 1976, S.75
und Österle, H.: Entwurf betrieblicher Informationssysteme. München 1981, S. 56f

79) vgl. Wedekind, H., Ortner, E.: Systematisches Konstruieren von Datenbankanwendungen, a.a.O., S.22f

80) vgl. Grochla, E., Garbe, H., Gillner, R.: Gestaltungskriterien für den Aufbau von Datenbanken, a.a.O., S.39

3.2.3 WAHL PASSENDER ENTSCHEIDUNGSKRITERIEN

Um Entscheidungskriterien für den Aufbau der Datenbank festlegen zu können, muß zuvor sowohl der Nutzen dieses Vorhabens als auch dessen Meßbarkeit (soweit dies möglich ist) definiert werden.
Die konkreten Ziele, die im vorangehenden Abschnitt definiert wurden, sind daraufhin zu überprüfen inwieweit sie operational sind. Sie sind gegebenenfalls in operationale Unterziele zu gliedern. Bei der Definition dieser Unterziele ist zu beachten, daß dadurch unter Umständen eine Suboptimierung vorgenommen wird, bei der das Endergebnis vom Gesamtoptimum zuweit entfernt ist. [81)]
Die für den speziellen Anwendungsfall passenden Entscheidungskriterien lassen sich aus den bereits definierten Zielen und ihren Unterzielen ableiten. So gefundene Entscheidungskriterien sind auf den Grad ihrer Unsicherheit zu untersuchen und entsprechend ihrem Nutzen zu gewichten.

---

81) vgl. Hanssmann, F.: Einführung in die Systemforschung, a.a.O., S.36

3.2.4 FORMULIERUNG DER ENTSCHEIDUNGSALTERNATIVEN

Entscheidungsalternativen [82] gibt es zunächst hinsichtlich der

- Zielsetzung

hinsichtlich des Benutzers bei der

- Gestaltung der Schnittstellen
- Anpassung an das Benutzerverhalten
- Anpassung hinsichtlich des Informationsangebots
- Anpassung hinsichtlich des Informationsbedarfs

hinsichtlich der Daten bezüglich der

- Herkunft des Datenbestandes
- Variabilität der Daten
- Relation der Daten zu zeitlichen Bezugspunkten

Die Entscheidungsalternativen können demnach erzeugt werden [83]

- aus den Aufgaben der Organisation
- durch Lockerung der Randbedingungen
- durch Verbesserung bereits bekannter Alternativen
- durch Veränderung zeitlicher Faktoren
  (z.B. Realisierungsdauer)

Wie die Entscheidungsalternativen die Struktur der logischen Datenbank beeinflussen können zeigt Bubenko in einem Ablaufplan. [84] So können beispielsweise die Ziele, Struktur und Aktivitäten eines Bereichs sowohl die Struktur und Eigenschaften der Objekte der Datenbank beeinflussen, als auch die Ein- und Ausgabetypen und die Häufigkeit des Datenbankzugriffs. Diese können wiederum ihrerseits die Definition der Informationsstruktur und ihrer Relationen beeinflussen und so fort.

---

82) vgl. Grochla, E., Garbe, H., Gillner, R: Gestaltungskriterien für den Aufbau von Datenbanken, a.a.O.,S.26ff

83) vgl. Hanssmann, F.: Einführung in die Systemforschung, a.a.O., S.56ff

84) vgl. Bubenko, J.A., et al: From Information Requirements to DBTG-Data Structures, a.a.O., S.75
und Berild, S.: CS4 - A Tool for Database Design by Infological Simulation. In: Proc. of the 3rd Int. Conf. on Very Large Data Bases. New York 1977, S.533
und Teorey, T.J., Fry, J.P.: The Logical Record Access Approach to Database Design. Computing Surveys, 12 (1980), S. 184

3.2.5 AUSWAHL RELEVANTER UMWELTFAKTOREN

3.2.5.1 Analyse der entities und ihrer Ausprägungen

Unter Umweltfaktoren werden hier die Parameter eines Modells verstanden. [85] Es werden zielrelevante und handlungsrelevante Umweltfaktoren unterschieden. Die zielrelevanten Umweltfaktoren sind primär mit dem Modell (und seiner Zielfunktion) verbunden, während die handlungsrelevanten Umweltfaktoren den Entscheidungsalternativen zuzuordnen sind. Diese Umweltfaktoren können nach inhaltlichen, technisch ökonomischen und zeitlichen Gesichtspunkten, sowie nach ihrem Bestimmtheitsgrad unterschieden werden.
Es empfielt sich daher eine Übersicht aller Umweltfaktoren aufzustellen, die die verbalen und symbolischen Bezeichnungen, die Maßeinheit und den Bestimmtheitsgrad der Umweltfaktoren und sonstige charakteristische Merkmale enthält. Beispielsweise kann man für alle Attribute folgende Angaben machen: [86]

- Name
- Synonyme
- Identifikation
- Häufigkeit (Mengengerüst)
- Herkunft

Zu den charakteristischen Merkmalen von Informationen gehören:

- charakteristische Ausprägungen der Datenelemente
- typische Abhängigkeiten der Datenelemente

---

85) vgl. Hanssmann, F.: Einführung in die Systemforschung, a.a.O., S.70 ff

86) vgl. Davenport, R.A.: Data Analysis - Experience With a Formal Methodology. EURO-IVIP 1979: Proc. of the European Conference on Applied Inf. Technology, hrsg. von P.A. Samet. Amsterdam 1979, S.60
und Kahn, B.: A Structured Logical Database Design Methodology, a.a.O., S.17
und Bubenko jr., J. A.: IAM: Inferential Abstract Modeling Approach to Design of Conceptual Schema. ACM SIGMOD Int. Conf. on Management of Data. New York 1977, S. 66f
und Mitoma, M.F. Irani, K.B.: Automatic Data Base Schema Design and Optimization. Data Base Design Methodology, an Intensive Short Course, Engineering Summer Conf. Ann Arbor 1976, S. 290 ff
und Bubenko, jr., J.A.: Data Models and their Semantics, a.a.O., S. 122 ff
und Teorey, T.J., Fry, J.P.: The Logical Record Access Approach to Database Design, a.a.O., S. 194 ff

Zur Bestimmung der Umweltfaktoren - also der Datenelemente - können folgende Fragestellungen beitragen: [87]

- wessen Bedürfnisse (außerhalb der Organisation) sucht die Organisation zu befriedigen
- wie werden diese Bedürfnisse der Organisation mitgeteilt
- wie werden diese Informationen festgehalten und innerhalb der Organisation weitergeleitet
- an welchen Stellen des Informationsflusses werden Entscheidungen ausgelöst und nach welchen Regeln
- wie werden diese Entscheidungen den ausführenden Stellen mitgeteilt - wie verläuft der Informationsfluß
- welche Hilfsmittel (Ressourcen) werden durch den Informationsfluß in Bewegung gesetzt

Durch die Beantwortung dieser Fragen gewinnt man nicht nur Erkenntnisse, welche Datenelemente angesprochen werden. Vielmehr kann man dadurch auch bestimmen, welche Datenelemente über andere Datenelemente angesprochen werden und somit die funktionalen Abhängigkeiten festlegen.
Um sich die Zusammenhänge zwischen den Datenelementen klarzumachen, kann man die Bachmann-Diagramme verwenden [88], aus denen die Abhängigkeit der Datenelemente untereinander ersichtlich wird.
In der Anfangsphase der Analyse der entities wird mit Datenelementen gearbeitet, die selbst noch Relationen sind. Durch Aggregation und Generalisierung [89] können die Datenelemente schrittweise detaillierter definiert werden, bis letzlich entities als kleinste Einheit der Datenbank erreicht werden. Durch Typisierung und Detaillierung der Datenelemente wird automatisch eine semantische Integrität der Daten erreicht. [90]
Nach Davenport [91] kann die Datenanalyse in die beiden Teile entity Analyse und Funktional-Analyse gesplittet werden. Die Funktional-Analyse wiederum läßt sich in die Bestimmung der m:n-Verbindungen und die Festlegung der Elementaren Funktionalrelationen gliedern.

---

87) vgl. Hanssmann, F.: Einführung in die Systemforschung, a.a.O., S. 30
88) vgl. Bachman, C.W.: Data Structure Diagrams. Database 1 (1969), S. 4-10
89) vgl. Smith, J.M., Smith, D.C.P.: Database abstractions: aggregation and generalization, a.a.O., S.105 ff
90) vgl. Smith, J.M., Smith, D.C.P.: Database abstraction, a.a.O., S. 25 ff
und Flory, A., Kouloumdjian, J.: A Model and a Method for Logical Database Design. In: IEEE Proc of the 4th Int. Conf. on Very Large Databases, o.O., 1978, S.333 ff
91) vgl. Davenport, R.A.: Data Analysis, a.a.O., S.54 ff

3.2.5.2 Analyse der funktionalen Abhängigkeiten

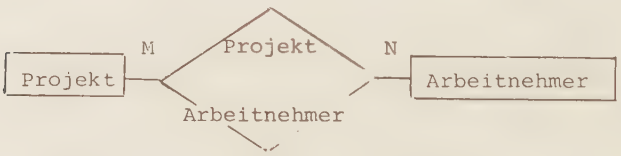
3.2.5.2.1 Bestimmung der m:n-Verbindungen

Um Elementare Funktional-Relationen (EFR) festlegen zu können, ist es zunächst erforderlich, die funktionalen Abhängigkeiten zu kennen. Da die Eigenschaften von entities ganz wesentlich durch ihre Attribut-Werte bestimmt werden, ist es von Bedeutung, wie die Attribut-Werte unterschiedlicher entities zueinander in Beziehung stehen. Von Bedeutung für die logische Strukturierung der Datenbank ist es, ob mehrere entities von einem entity funktional abhängig oder ob sie von mehreren entities funktional abhängig sind. Dabei unterscheidet man 1:1, 1:n und m:n Verbindungen.

Bei 1:1 Verbindungen ist das Problem einfach.

Für 1:n und m:n Verbindungen ist es ratsam, entity-relationship-Diagramme (E-R-Diagramme) zu zeichnen, damit man einen Überblick über die Relations-Typen erhält. [92)]

Beispiel:



entity-set relationship-set entity-set

Man kann diese Beziehungen auch in einer Matrix darstellen, die dann den Vorteil hat, daß sie maschinell auswertbar ist.

---

92) vgl. Chen, P. P.-S.: The Entitiy-Relationship Model - Basis for Enterprise View of Data. In: AFIPS Conf. Proc., hrsg. von R.R. Korfhage. Montvale, New Jersey 1977, S. 79
und Adiba, M., et al: A Unified Approach for Modelling Data Base Design. In: Modelling in Data Base Management Systems, hrsg. von G,M.Nijssen. Amsterdam 1976, S.313ff
und Teorey, T.J., Fry, J.P.:Logical Database Design: A Pragmatic Approach. In: INFOTECH State of the Art Report, Data Base Technology. Maidenhead 1978, S.357 ff
und ADV-ORGA (Hrsg.): Seminarunterlagen, a.a.O., S. 3.5.100 f
und Martens, J.-M.: Anwendungsbezogene Datenorganisation formatierter Datenbanken. München 1979, S. 92 ff

Führt man in den Zeilen und Spalten alle entity-sets oder entities auf, so kann man in den Matrixplätzen die Art der Verbindung eintragen. Die Werte der Hauptdiagonalen sind in dieser Matrix nicht belegt, bzw. haben immer den Wert 1.
Beispiel: an einem Projekt seien maximal 20 Mitarbeiter beschäftigt und jeder Mitarbeiter arbeitet an (beziehungsweise betreut) maximal 3 Projekte. Jedes Projekt hat einen Projektleiter, der maximal 2 Projekte betreut.

m \ n	Projekt	Projektleiter	Mitarbeiter
Projekt		1	20
Projekt-leiter	2		
Mitarb.	3		

Dann können m:n Verbindungen in der Weise aufgelöst werden, daß Hilfsknoten [93] eingefügt werden und dadurch m:1 und 1:n Verbindungen entstehen. Eine derartige Auflösung könnte so aussehen, daß statt einer direkten Zuordnung der Attribute eine Adresse einer weiteren Tabelle als Hilfsknoten angegeben wird.
Als weiteres Hilfsmittel zur Bestimmung der m:n-Verbindungen schlagen Chen und Nijssen [94] eine tabellarische Liste aller möglichen Attributwerte eines entity-sets vor.
Dadurch gewinnt man einen Überblick darüber, welche entities zu einem entity-set gehören und welche Werte diese entities annehmen können.

---

93) vgl. Vetter, M.: Data Base Design by Applied Data Synthesis, a.a.O., S.431f
und Zaniolo, C., Melkanoff, M.A.: On the Design of Relational Database Schemata, a.a.O., S.43f

94) vgl. Chen, P. P.-S.: The Entity-Relationship-Model - Toward a Unified View of Data. ACM Trans. on Database Systems 1 (1976), S.9-36
und Nijssen, G.M.: A Framework for Advanced Mass Storage Applications. In: Medinfo 80, Proc of 3rd Conf. on Medical Informatics. Amsterdam 1980
und Falkenberg, E., Hahn, H.: Information Analysis of Power Distribution Networks. Siemens Forsch.- und Entwicklungsbericht. Bd. 10 Berlin 1981, S.246 ff
und Raver, N., Hubbard, G.U.: Automated Logical Data Base Design: Concepts and Applications. Informatie 20 (1978), S. 355 ff

Die Liste der Attributwerte wird man jedoch nur beispielhaft führen können. Zum einen sind die möglichen Werte zum Zeitpunkt des Datenbankaufbaus nicht alle bekannt, zum anderen würde diese Liste viel zu umfangreich werden. Durch eine derartige Liste können auch semantische Fehler festgestellt werden, beispielsweise ob derselbe Attribut-Wert für unterschiedliche entities vergeben wurde, was zwar berechtigt sein kann, aber nicht immer berechtigt sein muß.
Neben den Attributwerten sollten auch die Eigenschaften der Attribute bestimmt werden, da auch daraus u.U. funktionale Abhängigkeiten gebildet werden können. [95)]
Jeder entity-Typ soll mit allen anderen entity-Typen in Zusammenhang gebracht werden, damit geprüft werden kann, ob sich daraus später eine sinnvolle Fragestellung ergeben kann und eine funktionale Abhängigkeit definiert werden kann. [96)]

---

95) vgl. Baldissera, C., Ceri, S., Pelagatti, G., Bracchi, G.: Interactive Specification and Formal Verification of User's Views in Database Design. In: Proc 5th. Int. Conf. on Very Large Data Bases. Rio de Janeiro 1979, S. 266 ff
und Pfiffner, M.: Mit Methode zu integrierten Datenbanken. Output, 9 (1980), Heft 1, S. 20 f

96) vgl. Davenport, R.A.: Data Analysis - Experience with a Formal Methodology, a.a.O., S. 57
und Davenport, R.A.: Data Analysis for Database Design. The Australian Computer Journal, 10 (1978), 4, S. 125 f
und Wenger, D.: Ein methodischer Weg zur konzeptionellen Datenbank. Output, 10 (1981), Heft 1, S. 24 ff

3.2.5.2.2 Bestimmung der Elementaren Funktionalrelationen

Aus der Analyse der entities, ihrer Attribute und der Bestimmung der m:n Verbindungen können die Elementaren Funktionalrelationen bestimmt werden. Unter Elementaren Funktionalrelationen versteht man Relationen, deren Elemente voll funktional abhängig vom Schlüssel sind. [97] Um volle funktionale Abhängigkeit ermitteln zu können, ist als Voraussetzung festzustellen, welche entities identifizierende Eigenschaften haben und welche entities dadurch identifiziert werden. Entities mit identifizierender Eigenschaft sind Schlüsselkandidaten oder Elemente eines Schlüsselkandidaten.
Erste Anhaltspunkte dafür liefert die Analyse der m:n Verbindungen. Die Matrix der m:n Verbindungen kann insoweit zur Bildung der EFR herangezogen werden, als dort zeilenweise die Schlüsselkandidaten gekennzeichnet werden können. Bei 1:m Verbindungen kann man die zu untersuchenden Relationen zeilenweise aus der Matrix ablesen. Alle m:n Verbindungen sind zuvor durch Einfügen eines Hilfsknotens - wie bereits erwähnt - aufzulösen.
Werden nun alle Schlüsselelemente und die vom jeweiligen Schlüssel voll funktional abhängigen Nichtschlüsselelemente zusammengefaßt, so erhält man Elementare Funktionalrelationen. Zweckmäßigerweise werden in diesen Relationen die Schlüsselelemente links angeordnet und die Nichtschlüsselelemente rechts davon.
Die so gefundenen EFR sind nach Schlüsselelementen zu sortieren. Danach lassen sich solche Relationen zusammenfassen, die dieselben Schlüsselelemente haben. Sodann ist der Einsatz vorbereitender Algorithmen möglich, wie sie

---

97) vgl. Delobel, C., Casey, R.G.: Decomposition of a Data Base and the Theorey of Boolean Switching Functions. IBM Journal of Research and Development 17 (1973), S. 374 - 386
und Wedekind, H.: Datenbanksysteme I, a.a.O., S. 71f

beispielsweise von Hubbard oder Bernstein oder Mijares, Peebles beschrieben werden.[98] Algorithmen zur Normalisierung von Relationen bei zusammengesetzten Schlüsselkandidaten - die hier ausgeklammert werden - findet man bei Bernstein und anderen.[99]

Für alle Schlüsselelemente sind die voraussichtlichen Häufigkeiten des Datenzugriffs zu schätzen. Die EFR sind in einer Adjazenzmatrix[100] für das Normalisierungsverfahren vorzugeben. Dabei sollten die entities in Zeile/Spalte in der Reihenfolge der Häufigkeit des Datenzugriffs vorgegeben werden. Der Grund dafür wird nachfolgend noch ausführlich beschrieben.

---

98) vgl. Hubbard, G.U.: Techniques for Automated Logical File Design. NYU Symposium of Database Design. New York 1978, S. 88
und Bernstein, P.A.: Normalization and Functional Dependencies in the Relational Database Model, a.a.O.
und Mijares, O., Peebles, R.: A Methodology for the Design of Logical Data Base Structure. In: Modelling in Data Base Management Systems, hrsg. von G.M. Nijssen. Amsterdam 1976, S.25-47

99) vgl. Bernstein, P.A.: ebenda
und Fadous, R.Y., Forsyth,J.: Finding Candidate Keys for Relational Data Bases. In: Proc. ACM SIGMOD intern. Conf. on Management of Data, San José Calif., hrsg. von W.F. Kind. New York 1975, S. 203 ff
und Ullman, J.D.: Principles of Database Systems. Comp. Science Press 1980, S. 194 ff
und Adiba, M., et al: A unified Approach for Modelling Data Base Design, a.a.O., S.328 ff

100) vgl. Meyer, M., Hansen, K., Rohde, M.: Mathematische Planungsverfahren I. Essen 1973, S.106:
Eine Adjazenzmatrix gibt die Struktur eines Graphen dadurch wieder, daß ihre Zeilen und Spalten den Knoten des Graphen zugeordnet sind und ihre Elemente den Wert 1 haben, sofern zwischen den betreffenden Knoten eine gerichtete Kante - also eine voll funktionale Abhängigkeit - existiert ansonsten den Wert null.

3.3 MODELLKONSTRUKTION

3.3.1 ERZEUGEN VON RELATIONEN IN OPTIMALER 3NF

Gesucht wird ein rechenzeitgünstiges Verfahren, das Relationen in optimaler 3NF liefert. Die Forderung nach optimaler 3NF wird in der Literatur damit begründet, daß dadurch Anormalitäten beim Änderungsdienst der Datenbank vermieden werden.
Um den Verwaltungsaufwand im Änderungsdienst der Datenbank so gering wie möglich zu halten, sollen die funktionalen Abhängigkeiten soweit reduziert werden, daß jedes entity durch genau eine funktionale Abhängigkeit bestimmt wird.
Das bedeutet gleichzeitig, daß die Lösung keine transitiven Abhängigkeiten aufweist und keine Zyklen (auch keine inversen Abhängigkeiten) enthält. Damit die Auskunftsbereitschaft der Datenbank nicht leidet, müssen alle entities die voll funktional voneinander abhängig sind auch in der Lösung untereinander erreichbar sein.
Zur Auswahl standen zwei große Gruppen von Verfahren:

- Algorithmen zur Bestimmung von minimalen Überdeckungen
- Baumalgorithmen

Algorithmen zur Bestimmung von minimalen Überdeckungen wurden aus folgenden Gründen nicht gewählt:

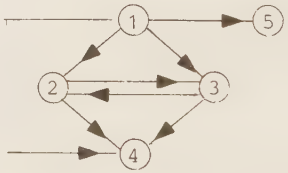
- sie sind rechenzeitaufwendig
- enthält der Graph aus funktionalen Abhängigkeiten Zyklen, so werden diese Verfahren rechentechnisch sehr aufwendig
- das Ergebnis einer minimalen Überdeckung kann ein Zyklus sein [101)]

Für das neu zu entwickelnde Verfahren wurde ein Baumalgorithmus gewählt, dieser liefert

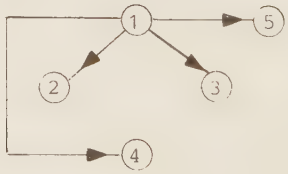
---

101) vgl. HSU, H.T.: An Algorithm for Finding a Minimal Equivalent Graph of a Digraph. Journal of the ACM 22 (1975), S.11 ff

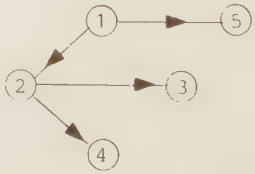
immer Relationen in 3NF. Sie lassen weder transitive noch pseudotransitive Abhängigkeiten zu. Ein Baum liegt dann vor, wenn in einem zusammenhängenden Graphen jeweils zwei Knoten durch genau einen Weg verbunden sind. Ein Baum mit n Knoten enthält immer n-1 Kanten. [102] Da spannende Bäume, denen die kürzeste Entfernung von der Wurzel zu den einzelnen Knoten zugrundeliegt, zyklenunempfindlich sind, wäre es wünschenswert, einen solchen Algorithmus einsetzen zu können.
Das würde jedoch bei dem Beispiel nach Wedekind (vgl. 2.3.2)



zu folgendem Ergebnis führen:



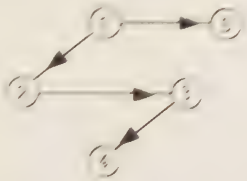
Diese Relationen wären zwar in 3NF, der Knoten 3 käme als Schlüsselkandidat aber nicht mehr in Frage. Auch bei folgender Lösung: [103]



---

102) vgl. Dörfler, W., Mühlbacher, J.: Graphentheorie für Informatiker. Berlin 1973, S.23
103) vgl. Codd, E.F.: Further Normalization of the Data Base Relational Model, a.a.O., S. 49

Wäre eine eventuelle spätere Anfrage an die Datenbank mit den Knoten 3 als Schlüssel nicht mehr möglich, es würden also Informationen verlorengehen.
Gesucht wäre demnach folgende Lösung:



Wird einem spannenden Baum nicht die kürzeste Entfernung zwischen Wurzel und den Knoten zugrundegelegt, sondern die maximale Entfernung, so kann obige Lösung erreicht werden. Für das neu zu entwickelnde Verfahren wurde deshalb ein Baumalgorithmus mit den maximalen Entfernungen zwischen den Knoten gewählt.
Einschränkend muß jedoch gesagt werden, daß die Ermittlung der maximalen Entfernung von der Wurzel zu den Knoten voraussetzt, daß es sich um einen zyklenfreien Graphen handelt, da sonst die maximale Entfernung immer unendlich ist. Hat ein Graph Zyklen, so müssen diese zuvor entfernt werden (d. h. eine Kante im Zyklus muß eliminiert werden), da es kein Verfahren gibt, das Zyklen ohne mehrmaliges Durchlaufen der Kantenzüge auf einfache Art erkennt. Daß die Existenz von Zyklen - insbesondere in der Lösung - das Ergebnis stören kann, wurde eingangs bereits erwähnt und wird in einem späteren Abschnitt noch gezeigt.
Es wurde nun nach einem Baumalgorithmus gesucht, der möglichst rechenzeitgünstig ist. Als von d'Esopo entwickeltes Verfahren zur Bestimmung der kürzesten Verbindungen zwischen festen Knoten [104], wurde in einer Variation von Pape [105] als besonders rechenzeitgünstig

104) vgl. Pollack, M., Wiebenson, W.: Solution of the Shortest Route Problem - a Review. Operations Research 8 (1969), S.224 ff
105) vgl. Pape, U.: Kürzeste Wege in Netzwerken zwischen festen Knoten. In: Veröff. des Inst. für Stadtbauwesen der Techn. Universität Braunschweig. Heft 7 , Braunschweig 1971, S.52

erkannt. Da es auch verhältnismäßig wenig Speicherplatz benötigt, scheint es das derzeit wirtschaftlichste Verfahren zur Bestimmung von Entfernungen zwischen festen Knoten zu sein. Dieses Verfahren wurde hier so modifiziert, daß es neben den minimalen auch die maximalen Entfernungen liefert und ferner so erweitert, daß es auch den zugehörigen spannenden Baum ermittelt. Somit ist gewährleistet, daß das neue Verfahren besonders rechenzeitgünstig ist. Da ein Baum - wie im Kapitel 3.6 noch gezeigt wird - immer Relationen in optimaler 3NF liefert, erscheint die Verwendung des Algorithmus nach d'Esopo als wirtschaftlichste Lösung zur Bestimmung eines spannenden Baumes und damit zur Ermittlung von Relationen in optimaler 3NF.

3.3.2 MINIMIERUNG DER ANZAHL DER RELATIONEN

Die gesuchten Relationen in 3NF sollen in dem Sinn optimal sein, daß es keine andere Anordnung mit weniger Relationen gibt. [106] Bei der Bestimmung der voll funktionalen Abhängigkeiten wurde bereits gesagt, daß alle diejenigen Knoten des Graphen - der aus voll funktionalen Abhängigkeiten gebildet werden kann - Schlüsselkandidaten sind, wenn von diesen Knoten Pfeile wegführen. Zum Zeitpunkt der Konzeption der logischen Datenbankstruktur können nicht alle künftigen Anfragen an die Datenbank bekannt sein. Deshalb würde eine Einschränkung der Anzahl der Schlüsselkandidaten immer einem Verlust an Information gleichkommen. Aus diesem Grund soll durch den Normalisierungsprozeß die Zahl der Schlüsselkandidaten nicht verringert werden. Das bedeutet dann aber auch, daß die Anzahl der gesuchten Relationen in 3NF durch die Zahl der Schlüsselkandidaten bestimmt wird.

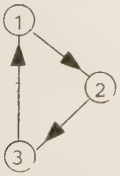
---

106) vgl. Wedekind, H.: Datenbanksysteme I, a.a.O., S. 67

3.3.3 AUSSCHLIEßEN VON ZYKLEN

Transitive Abhängigkeiten werden in der 3NF ausgeschlossen, weil sie zu Anormalitäten im Änderungsdienst der Datenbank führen können.

Bei obigen funktionalen Abhängigkeiten wird der Wert von 2 und 3 durch den Wert von 1 bestimmt. Der Wert von 3 außerdem durch den Wert von 2. Um keine doppelten Abhängigkeiten zu schaffen, die im Änderungsdienst alle abgehandelt werden müßten, kann und muß auf die funktionale Abhängigkeit von 1-3 verzichtet werden, da sie denselben Wert liefern muß wie 1-2-3.
Liegt nun ein Zyklus vor:



so bedeutet dies, daß im Änderungsdienst - bei einer Änderung von 1 - die Attribute 2 und dadurch auch 3 geändert werden müssen. [107] Durch die funktionale Abhängigkeit 3-1 muß - bei Änderung des Attributs 3 - auch das Attribut 1 geändert werden und so fort.
Damit wäre wohl bereits eine Schleife im Änderungsdienst der Datenbank vorprogrammiert.
Reduziert man diesen Zyklus, so stellt man fest, daß eine volle funktionale Abhängigkeit des Attributs 1 (oder 2 oder 3) von sich selbst aufgebaut wurde.
Es erscheint wenig sinnvoll eine solche Abhängigkeit in der Lösung zuzulassen.
Daher muß man bei der Auflösung von Zyklen die funktionale Abhängigkeit eliminieren, die am unwichtigsten ist.

---

107) vgl. Wedekind, H.: Datenbanksysteme I, a.a.O., S.57

3.3.4 BERÜCKSICHTIGUNG DER HÄUFIGKEIT DES DATENZUGRIFFS

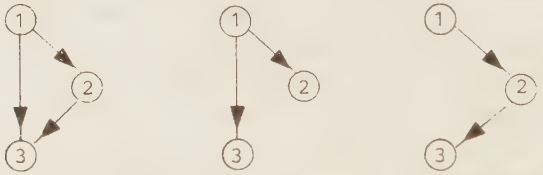
3.3.4.1 Mögliche Alternativen

Für die Berücksichtigung der Häufigkeit des Datenzugriffs wurden zwei Alternativen untersucht:

- Bewertung der Kanten des Graphen
- Bewertung der Knoten des Graphen

3.3.4.2 Bewertung der Kanten des Graphen

Aus folgendem Grund wurde dem Baumalgorithmus die maximale Entfernung zwischen den Knoten zugrundegelegt:



wird - nun zur Auflösung der transitiven Zusammensetzungen, im linken Bild - die Kante 2-3 entfernt, so kommt das Attribut 2 als Schlüssel nicht mehr in Frage. Mit dem Schlüssel 2 kann auch keine Anfrage beantwortet werden, bei der als Antwort das Attribut 3 erwartet würde. Derselbe Verlust an Information tritt auf, wenn die Kante 1-2 entfernt wird. Wird dagegen die Kante 1-3 entfernt, so können alle Anfragen beantwortet werden, die Erreichbarkeit der Knoten untereinander ist also erhalten. Diese Lösung wird mit einem Baumalgorithmus erreicht, sofern die Kanten gleich bewertet werden und nach der maximalen Entfernung zwischen den Knoten gesucht wird. Würde nun die Häufigkeit des Datenzugriffs, durch die Bewertung der Kanten berücksichtigt, so könnte eine Kante so ungünstig bewertet werden, daß ein Verlust an Information eintritt, wie er oben beschrieben wurde.

Beispiel:

10,11,25 = Häufigkeit des Datenzugriffs

Die Distanz 21 (1-2-3) ist geringer als 25 (1-3), weshalb der Ast 1-3 und 1-2 in den Baum genommen würde. Diese Lösung wäre jedoch ungünstig, wie oben bereits beschrieben wurde. Aus diesem Grund wird die Berücksichtigung der Häufigkeit des Datenzugriffs nicht durch die Bewertung der Kanten des Graphen vorgenommen.

3.3.4.3 Bewertung der Knoten des Graphen

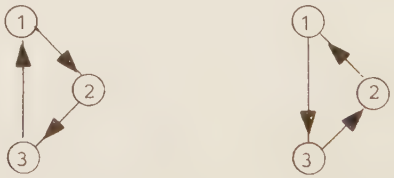
3.3.4.3.1 Einfluß der Knotennumerierung auf den Baumalgorithmus

Der verwendete Baumalgorithmus ist - mit einer Ausnahme - unabhängig von der gewählten Knotennumerierung. Das Programm geht lediglich davon aus, daß der Graph in Form einer Adjazenzmatrix vorgegeben wird, die so interpretiert wird, daß die Zeile/Spalte der Knotennummer entspricht.
Lediglich dann, wenn die Distanz von der Wurzel zu einem bestimmten Knoten - über zwei unterschiedliche Pfade - gleich ist, dann entscheidet die niedrigere Knotennummer des Vorgängerknotens welche Kante in den Baum genommen wird.
Beispiel:

unter der Voraussetzung, daß die Kanten gleich bewertet werden, wird die Kante 2-4 in den Baum genommen, da 2 die niedrigere Knotennummer als 3 hat.

3.3.4.3.2 Einfluß der Knotennumerierung auf die Zyklenauflösung

Werden maximale Entfernungen eines zusammenhängenden Graphen zwischen festen Knoten bestimmt und enthält der Graph Zyklen, so sind die maximalen Entfernungen immer unendlich. Da es - wie bereits erwähnt - kein Verfahren gibt, das Zyklen ohne mehrmaliges Durchlaufen der Kantenfolgen erkennt - müssen Zyklen zuvor aufgelöst werden. Das Programm verwendet zum Auffinden von Zyklen einen Algorithmus von Wille, Gewald und Weber.[108)]
Die Knotennumerierung des Graphen ist für das Auffinden - nicht jedoch für das Auflösen - von Zyklen unerheblich. Das Verfahren zur Zyklensuche wurde hier so modifiziert, daß die unwichtigste Kante im Zyklus eliminiert wird. Es wird angenommen, daß die Knoten des Graphen in der Reihenfolge der Häufigkeit - in der sie Schlüssel sind - durchnumeriert werden. Das bedeutet, daß der wichtigste Schlüssel die Knotennummer 1 erhält. Unter dieser Voraussetzung ist die unwichtigste Kante - bei einem Zyklus im Uhrzeigersinn - die Kante von der höchsten Knotennummer zur niedrigsten.
Beispiel:



Im linken Bild wäre die Kante 3-1 die unwichtigste. Bei einem Zyklus der gegen den Uhrzeigersinn läuft, wie im rechten Bild, wäre die unwichtigste Kante die Kante 3-2.

---

108) vgl. Wille, H., Gewald, K., Weber, H.: Netzplantechnik - Methoden zur Planung und Überwachung von Projekten. Bd. I. München 1972, S.147 ff

3.3.4.3.3 Unerwünschte Beeinträchtigung der Lösung durch das Entfernen einer bestimmten Kante im Zyklus

Es kann bei dem vorgesehenen Verfahren vorkommen, daß die Erreichbarkeit der Knoten untereinander dadurch leidet, daß eine bestimmte Kante im Zyklus des Graphen entfernt wurde. Das Programm arbeitet im automatischen Ablauf so, daß es als unwichtigste Kante die Kante von der höchsten Knotennummer zur niedrigsten Knotennummer (sofern diese Kante existiert, ansonsten die Kante von der höchsten Knotennummer zur nächstniedrigeren Knotennummer) entfernt. Steuert der Anwender den Ablauf der einzelnen Programmschritte selbst, so hat er die Möglichkeit anzugeben welche Kante im Zyklus eliminiert werden soll. Enthält der Graph nur einen Zyklus (aus 3 oder 4 Kanten), so ist es relativ einfach diejenige Kante zu finden, die eliminiert werden kann ohne die Erreichbarkeit der Knoten untereinander zu stören. Enthält ein Graph jedoch mehrere Zyklen, so ist eine solche Vorgehensweise nicht mehr sinnvoll. Daher wird im Kapitel 3.6 ein Weg gezeigt, wie man Zyklen so auflösen kann, daß die maximale Erreichbarkeit der Knoten untereinander immer gegeben ist.

3.3.4.4 Zusammenfassung

Werden die Knoten in der Reihenfolge der Häufigkeit des Datenzugriffs numeriert und erhält der am häufigsten angesprochene Schlüssel die Nummer 1 und das am seltensten angesprochene Attribut die höchste Knotennummer, so kann man dadurch folgendermaßen Einfluß auf das Ergebnis nehmen:

1. bei gleichen Distanzen wird die wichtigere Kante als Ast in den Baum genommen (vgl. 3.3.4.3.1)
2. enthält der Graph Zyklen, so wird die unwichtigste Kante eliminiert um so den Zyklus aufzulösen.

Anstelle der Knotennumerierung in der Reihenfolge der Häufigkeit des Datenzugriffs, wäre es auch denkbar eine Häufigkeitsmatrix bzw. einen Häufigkeitsvektor für die Knoten vorzugeben. Dies würde jedoch für das Einlesen, Protokollieren und Abarbeiten der Häufigkeitsmatrix oder des Vektors zusätzliche Rechenzeit kosten.
Programmtechnisch würde dennoch nur die Reihenfolge der Häufigkeiten berücksichtigt. Aus diesem Grund wird auf eine Häufigkeitsmatrix verzichtet.

3.3.5 MASCHINELLE ÜBERPRÜFUNG DER ERGEBNISSE

Es wurde festgestellt, daß Baumalgorithmen zwar immer Relationen in 3NF liefern, daß aber ein Verlust an Information auftreten kann, wenn Schlüsselkandidaten in der Lösung nicht mehr Schlüssel sind.
Dieser Fall wurde als mangelnde Erreichbarkeit der Knoten untereinander bezeichnet.
Die Ergebnisse wurden deshalb anhand einer Erreichbarkeitsmatrix überprüft. Diese Erreichbarkeitsmatrix wurde durch Potenzieren der Adjazenzmatrix des Baumes gefunden. [109)]
Hat ein Graph Zyklen, so sind zwei Fragen interessant:

1. Ist die Erreichbarkeit in Bezug auf den zyklenfreien Graphen (die Zyklen wurden aufgelöst) noch gegeben?
2. Ist die Erreichbarkeit in Bezug auf den ursprünglichen Graphen mit Zyklen noch gegeben?

Die erste Frage sollte immer positiv beantwortet werden können. Die Antwort auf die zweite Frage darf nur die Kanten betreffen, die in den Zyklen eliminiert wurden.
Für den Fall, daß die erste Frage eine mangelnde Erreichbarkeit ergibt, muß eine andere Kante im Zyklus eliminiert werden.
Liefert auch diese Vorgehensweise kein zufriedenstellendes Ergebnis so müssen die funktionalen Abhängigkeiten auf Vollständigkeit überprüft werden.
Es wird im Kapitel 3.6 noch gezeigt, daß die Summe der maximalen Distanzen ein weiteres Kriterium für die Qualität des Ergebnisses ist. Bei den berechneten Beispielen wurde festgestellt, daß die Summe der maximalen Distanzen immer dann den maximalen Wert aufweist, wenn die Erreichbarkeit der Knoten untereinander gegeben ist.

---

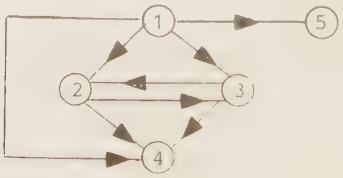
109) vgl. Neumann, K.: Operations Research Verfahren Bd.III, a.a.O., S.48 ff

3.4 MODELLVALIDIERUNG ANHAND VON BEISPIELEN

3.4.1 ÜBERPRÜFUNG DES ERSTEN BEISPIELS

3.4.1.1 CHARAKTERISTISCHE MERKMALE DES ERSTEN BEISPIELS[110]

Um das in Kapitel 3.3 entwickelte Verfahren zu überprüfen, wurden mehrere Beispiele mit charakteristischen Merkmalen ausgewählt. Einige der hier betrachteten Beispiele sind bereits im zweiten Kapitel beschrieben worden. In die weitere Betrachtung werden diese Beispiele nur noch mit ihrer Knotennummer eingehen und nicht mehr mit ihren symbolischen Attributnamen.



Charakteristisch für das erste Beispiel ist die inverse funktionale Abhängigkeit zwischen Knoten 2 und 3.

---

110) vgl. Wedekind, H.: Datenbanksysteme I, a.a.O., S.75
und Codd, E.F.: Further Normization of the Data Base Relational Model, a.a.O., S.33 ff
und Kapitel 2.3.2

3.4.1.2 ANALYSE DER ERGEBNISSE DES ERSTEN BEISPIELS

3.4.1.2.1 Überprüfung der Ergebnisse auf optimale 3NF

Die Ergebnisse (vgl. Anhang 53 und 54) sind in 3NF, da keine transitiven Abhängigkeiten in der Lösung sind. Alle ursprünglichen Schlüsselkandidaten (vgl. Anhang 53 Knoten 1,2,3) sind auch in der Lösung Schlüssel. Die Anzahl der Relationen entspricht der Anzahl der Schlüsselkandidaten. Eine weitere Reduzierung - der Anzahl der Relationen - wäre nur durch Verlust an Informationen möglich. Da es in der Lösung keine redundanten Abhängigkeiten gibt, sind die gefundenen Relationen in optimaler 3NF.

3.4.1.2.2 Überprüfung der Erreichbarkeit der Knoten

3.4.1.2.2.1 Erreichbarkeit im Vergleich mit dem zyklenfreien Graphen

Unabhängig von der Knotennumerierung (vgl. Anhang 53 und 54) wurde die ursprüngliche Erreichbarkeit der Knoten untereinander erhalten.

3.4.1.2.2.2 Erreichbarkeit im Vergleich zum Graphen mit Zyklen

Die mangelde Erreichbarkeit beschränkt sich auf die Kante, die bei der Zyklensuche eliminiert wurde. Sollte die Annahme, daß Zyklen in der Lösung nicht vorkommen dürfen nicht zutreffen, so müßte diese Kante in die Lösung zusätzlich aufgenommen werden.

Entschließt man sich dazu, so muß selbstverständlich erneut geprüft werden, ob dadurch transitive Abhängigkeiten entstanden sind. In diesem Beispiel wäre das nicht der Fall. Wird die Kante 3-2 (vgl. Anhang 53) in die Lösung aufgenommen, so ändert sich lediglich die Relation R3. R3 wird dann R3 (3,2,4).
Dieses Ergebnis ist identisch mit der Lösung x53 nach Wedekind (vgl. Kapitel 2.3.2).

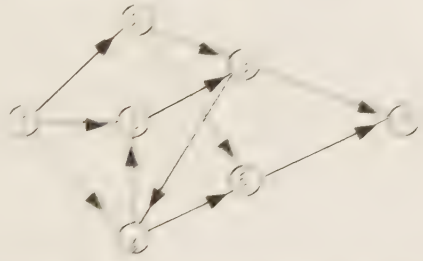
3.4.1.2.3 Berücksichtigung der Häufigkeit des Datenzugriffs und deren Auswirkung

Die Berücksichtigung der Häufigkeit des Datenzugriffs kann sich - wie bereits beschrieben - nur bei der Zyklenauflösung darauf auswirken, welche Kante im Zyklus eliminiert wird und bei der Bestimmung des Baumes bei zufällig gleichen Distanzen. Beide Fälle treffen in diesem Beispiel auf die Knoten 2 und 3 zu. Da dieses Beispiel mit Ausnahme des Knotens 5 symmetrisch ist, würde sich bei einer Vertauschung der Knoten 2 und 3 am Ergebnis nichts ändern. Die Berücksichtigung der Häufigkeit des Datenzugriffs durch eine andere Knotennumerierung (vgl. Anhang 54) hat für dieses Beispiel zu keiner Änderung der Lösung geführt.

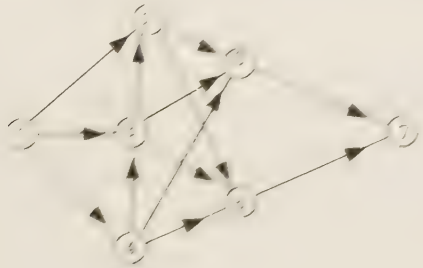
3.4.2 Behandlung des zweiten Beispiels

3.4.2.1 Graphentheoretische Methode des zweiten Beispiels[111]

Dieses Beispiel wird in zwei Varianten verwendet, die sich durch die Realisierung unterscheiden.
Zuerst wird der Fall betrachtet, daß sich durch die Realisierung ein Zyklus ergibt:



Das zweite leicht auszuwertende Beispiel enthält infolge gebildeter Realisierung keinen Zyklus mehr. Außerdem wurde die Kante 3-6 und 3-2 zusätzlich eingeführt sowie die Knotennummern 3 und 5 vertauscht:



---

(111) vgl. Ke, Te Chiang: Ganzzahlige Programmierung und Netzwerkfluß. Wiesbaden 1972, S.19

3.4.2.2 ANALYSE DER ERGEBNISSE DES ZWEITEN BEISPIELS

3.4.2.2.1 Überprüfung der Ergebnisse auf optimale 3NF

Die Ergebnisse in Anhang 55 bis 58 sind in 3NF. Die beiden Beispiele unterscheiden sich durch die Numerierung der Knoten 3 und 5. Durch diese unterschiedliche Numerierung wird jeweils eine andere Kante im Zyklus 3-4-5 entfernt. Die Lösung enthält jeweils 5 Relationen; in beiden Fällen kann die Anzahl der gefundenen Relationen nicht weiter reduziert werden ohne Verlust an Information in Kauf zu nehmen. Die gefundenen Relationen sind demnach in optimaler 3NF.
Die Ergebnisse für das abgewandelte zweite Beispiel (vgl. Anhang 59) sind zwar in 3NF, aber der Knoten 6 ist nicht mehr Schlüssel. Hier kann das Ergebnis nicht verbessert werden ohne transitive Abhängigkeiten aufzubauen. Nimmt man jedoch an, daß der Knoten 6 voll funktional vom Knoten 5 abhängig ist (vgl. Anhang 60), so bleiben alle ursprünglichen Schlüsselkandidaten in der Lösung Schlüssel. Das Ergebnis ist in optimaler 3NF.

3.4.2.2.2 Überprüfung der Erreichbarkeit der Knoten

3.4.2.2.2.1 Erreichbarkeit im Vergleich mit dem zyklenfreien Graphen

Im Beispiel im Anhang 55,56 ist die Erreichbarkeit der Knoten untereinander erhalten. In demselben Beispiel (vgl. Anhang 57 und 58) wurde durch eine geänderte Knotennumerierung eine andere Kante im Zyklus eliminiert. In diesem Fall gibt es keinen Baum der die ursprüngliche Erreichbarkeit der Knoten untereinander gewährleistet.

Beide Beispiele liefern die gleiche Anzahl von Relationen in 3NF. Dennoch ist das Ergebnis im Anhang 57 und 58 durch die mangelnde Erreichbarkeit der Knoten schlechter. Das abgewandelte zweite Beispiel (vgl. Anhang 59 und 60) enthält keine Zyklen. Bei dem Beispiel in Anhang 59 fehlt die Erreichbarkeit des Knotens 7 über den Knoten 6. Hier muß darauf verzichtet werden, da sonst eine transitive Verbindung aufgebaut würde (2-5-7 und 2-6-7). Bei der Überprüfung der funktionalen Abhängigkeiten fällt jedoch folgendes auf:wenn der Wert des Attributs 7 durch den Wert des Attributs 5 und durch den Wert des Attributs 6 bestimmt wird, dann müssen die Attribute 5 und 6 voll funktional abhängig voneinander sein. Aus diesem Grund wurde im Anhang 60 die volle funktionale Abhängigkeit des Knotens 6 von 5 aufgenommen. Dadurch kann ein Baum gefunden werden, der die Erreichbarkeit aller Knoten untereinander gewährleistet.

3.4.2.2.2.2 Erreichbarkeit im Vergleich zum Graphen mit Zyklen

Hier werden nur die Beispiele in Anhang 55 bis 58 betrachtet, da die Beispiele in Anhang 59 und 60 keine Zyklen enthalten. Die Erreichbarkeit der Knoten verringert sich - im Vergleich zur Erreichbarkeit des zyklenfreien Graphen - um diejenige Kante, die im Zyklus eliminiert wurde.

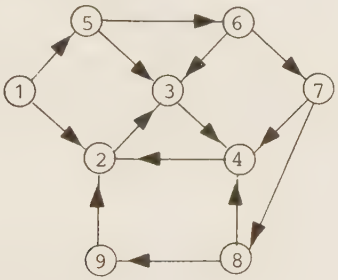
3.4.2.2.3 Berücksichtigung der Häufigkeit des Datenzugriffs und deren Auswirkung

Die Häufigkeit des Datenzugriffs wird durch die Reihenfolge der Knotennumerierung berücksichtigt, damit in Zyklen die unwichtigsten Kanten eliminiert werden.
Bei dem Beispiel in Anhang 55 und 56 wäre dies die Kante von Knoten 5 nach 3. Durch das Entfernen dieser Kante wird die Erreichbarkeit der Knoten untereinander nicht gestört.
Bei dem Beispiel in Anhang 57 und 58 wird jedoch über diese Kante häufiger zugegriffen, was durch Vertauschen der Knotennummern 5 und 3 zum Ausdruck gebracht wird.
Dadurch wird die Kante von Knoten 5 nach 4 eliminiert. Damit geht jedoch die Erreichbarkeit der Knoten untereinander zum Teil verloren. In diesem Fall muß der Graph nochmals überprüft werden um festzustellen, ob die Häufigkeit des Datenzugriffs tatsächlich Vorrang vor der Erreichbarkeit hat.
Bei den Beispielen in Anhang 59 und 60 ist die Häufigkeit des Datenzugriffs lediglich für die Knoten 5 und 6 interessant. Hier könnte eine Vertauschung der Knotennummern sinnvoll sein, wenn auf den Knoten 6 häufiger zugegriffen wird als auf den Knoten 5. Dann müßte jedoch auch die funktionale Abhängigkeit des alten Knotens 6 von 5 in die funtkionale Abhängigkeit des neuen Knotens 6 von 5 geändert werden. Da festgestellt wurde, daß Knoten 5 und 6 inverse Abhängigkeit aufweisen müssen, wäre dies auch durchaus möglich.

3.4.3 ÜBERPRÜFUNG DES DRITTEN BEISPIELS

3.4.3.1 CHARAKTERISTISCHE MERKMALE DES DRITTEN BEISPIELS [112)]

Dieses Beispiel enthält einen Zyklus und sehr viele transitive Abhängigkeiten. An diesem Besipiel wird später noch gezeigt, daß es nur eine Kante im Zyklus gibt, die eliminiert werden kann, ohne die Erreichbarkeit der Knoten untereinander zu stören.



3.4.3.2 ANALYSE DER ERGEBNISSE DES DRITTEN BEISPIELS

3.4.3.2.1 Überprüfung der Ergebnisse auf optimale 3NF

Die Ergebnisse in Anhang 62 und 64 sind in 3NF.
Die Ergebnisse in Anhang 64 sind auch in optimaler 3NF, da alle Schlüsselkandidaten erhalten wurden (Knoten 4 war durch die Zyklusauflösung nicht mehr Schlüsselkandidat). Die Ergebnisse im Anhang 64 liefern zwar eine geringere Anzahl von Relationen in 3NF, in der Lösung sind jedoch nicht mehr alle Schlüsselkandidaten als Schlüssel erhalten, insofern ist dieses Ergebnis schlechter.

112) Dieses Beispiel entstand in Diskussion mit Prof. Dr. H.-D. Ehrich, Universität Dortmund Abteilung Informatik I.
Es wurde einem Schreiben vom 20.11.1981 entnommen.

3.4.3.2.2 Überprüfung der Erreichbarkeit der Knoten

3.4.3.2.2.1 Erreichbarkeit im Vergleich mit dem zyklenfreien Graphen

Die Erreichbarkeit der Knoten - bei dem Beispiel in Anhang 61 und 62 - bleibt erhalten. Das Beispiel im Anhang 63 ist bis auf die Numerierung der Knoten 2-3-4 identisch mit dem Beispiel in Anhang 61. Durch die geänderte Numerierung der Knoten wird jedoch die neue Kante 4-2 (bisher 2-3) im Zyklus entfernt, was zu einer schlechteren Erreichbarkeit führt. Würde bei dem Beispiel im Anhang 61 - als dritte Alternative - die Kante 3-4 im Zyklus entfernt, so bliebe zwar die Erreichbarkeit nicht erhalten, wäre aber dennoch besser als die Erreichbarkeit bei dem Beispiel in Anhang 63 und 64.

3.4.3.2.2.2 Erreichbarkeit im Vergleich zum Graphen mit Zyklen

Die fehlende Erreichbarkeit wird jeweils um die Kante erhöht, die bei der Zyklenauflösung eliminiert wurde.

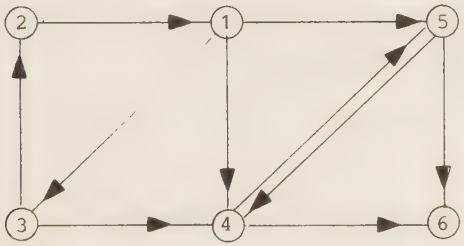
3.4.3.2.3 Berücksichtigung der Häufigkeit des Datenzugriffs und deren Auswirkung

Im vorliegenden dritten Beispiel (vgl. Anhang 61 bis 64) gibt es nur eine Reihenfolge der Knotennumerierung im Zyklus 2-3-4, die die ursprüngliche Erreichbarkeit der Knoten untereinander erhält (vgl. Anhang 61 und 62). Insoweit ist eine Berücksichtigung der Häufigkeit des Datenzugriffs in diesem Beispiel nur dann möglich, wenn trotzdem die Kante 4-2 (bezogen auf die Numerierung im Anhang 61) im Zyklus eliminiert wird.

3.4.4 ÜBERPRÜFUNG DES VIERTEN BEISPIELS

3.4.4.1 CHARAKTERISTISCHE MERKMALE DES VIERTEN BEISPIELS [113)]

Dieses Beispiel hat keine Wurzel und weist außerdem zwei Zyklen auf.



113) vgl. Neumann, K.: Operations Research Verfahren Bd. III, a.a.O., S. 89

3.4.4.2 ANALYSE DER ERGEBNISSE DES VIERTEN BEISPIELS

3.4.4.2.1 Überprüfung der Ergebnisse auf optimale 3NF

Die Ergebnisse des Beispiels im Anhang 65,66 sind in optimaler 3NF, da keine redundanten Abhängigkeiten enthalten sind.

3.4.2.2.2 Überprüfung der Erreichbarkeit der Knoten

3.4.2.2.2.1 Erreichbarkeit im Vergleich mit dem zyklenfreien Graphen

Die ursprüngliche Erreichbarkeit der Knoten untereinander wurde erhalten.

3.4.2.2.2.2 Erreichbarkeit im Vergleich zum Graphen mit Zyklen

Die fehlende Erreichbarkeit beschränkt sich auf die Kanten, die bei der Schleifenauflösung eliminiert wurden.

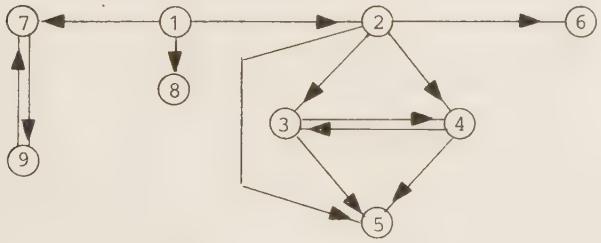
3.4.2.2.3 Berücksichtigung der Häufigkeit des Datenzugriffs und deren Auswirkung

Würde der Knoten 5 häufiger angesprochen als der Knoten 4 so müßten diese Knotennummern vertauscht werden (vgl. Anhang 67 und 68). Diese Lösung ist jedoch nur dann erstrebenswert, wenn tatsächlich der Zugriff über den Knoten 1 nach 4 und über 5 nach 6 so wichtig ist, daß auf den Zugriff 4-5 und 4-6 verzichtet werden kann.

3.4.5 ÜBERPRÜFUNG DES FÜNFTEN BEISPIELS

3.4.5.1 CHARAKTERISTISCHE MERKMALE DES FÜNFTEN BEISPIELS[114]

Für dieses Beispiel, das auch bereits im zweiten Kapitel betrachtet wurde, ist charakteristisch, daß es zwei inverse Abhängigkeiten aufweist und eine m:n Verbindung (7-8-2), die durch die Einführung eines Hilfsknotens (1) in eine m:1 und 1:n Verbindung aufgelöst wurde.



114) vgl. Vetter, M.: Data Base Design by Applied Data Synthesis, a.a.O., S.431

3.4.5.2 ANALYSE DER ERGEBNISSE DES FÜNFTEN BEISPIELS

3.4.5.2.1 Überprüfung der Ergebnisse auf optimale 3NF

Die gefundenen Relationen des fünften Beispiels (vgl. Anhang 69 bis 72) sind in optimaler 3NF, da alle Schlüsselkandidaten als Schlüssel erhalten und die voll funktionalen Abhängigkeiten redundanzfrei abgebildet wurden.

3.4.5.2.2 Überprüfung der Erreichbarkeit der Knoten

3.4.5.2.2.1 Erreichbarkeit im Vergleich mit dem zyklenfreien Graphen

Die ursprüngliche Erreichbarkeit der Knoten untereinander blieb erhalten.

3.4.5.2.2.2 Erreichbarkeit im Vergleich zm Graphen mit Zyklen

Hier fehlen im Vergleich zur ursprünglichen Erreichbarkeit genau die Kanten, die bei der Zyklenauflösung eliminiert wurden.

3.4.5.2.3 Berücksichtigung der Häufigkeit des Datenzugriffs und deren Auswirkung

Das fünfte Beispiel wurde mit zwei unterschiedlichen Knotennumerierungen berechnet (vgl. Anhang 69 und 71). Dabei konnte keine Auswirkung durch die Berücksichtigung der Häufigkeiten des Datenzugriffs auf die Lösung festgestellt werden.

3.5 DEDUKTION VON INFORMATIONEN AUS DEM MODELL

Wie vorangehend aufzuzeigen versucht wurde, liefert der Algorithmus immer Relationen in optimaler 3NF. Enthält der Graph nun funktionalen Abhängigkeiten Zyklen, so ist die Erreichbarkeit der Knoten untereinander genau davon abhängig, welche Kante zur Zyklenaufhebung eliminiert wurde. Die Erreichbarkeit von der Wurzel zu allen übrigen Knoten ist dagegen immer gegeben.
Die Anzahl der gefundenen Relationen in 3NF richtet sich nach der Anzahl der Schlüsselkandidaten. Werden die Schlüsselkandidaten reduziert, so geht im allgemeinen auch die Erreichbarkeit der Knoten untereinander verloren. Primär ist demnach wichtig, die ursprüngliche Erreichbarkeit der Knoten untereinander zu erhalten.
Festgestellt wurde, daß diese Erreichbarkeit immer dann gegeben war, wenn die Summe der maximalen Distanzen maximal war, auf diesen Zusammenhang wird im Kapitel 3.6 noch eingegangen.
Die Aufhebung von Zyklen - vor der Ermittlung der 3NF - ist erforderlich, da sonst der Wert der maximalen Distanz zwischen den Knoten unendlich wird.
Im Änderungsdienst einer Datenbank kann - ebenso wenig wie bei der Bestimmung der maximalen Distanz zwischen Knoten - festgestellt werden, ob die Ursache für eine mehrmalige Änderung eines Attributs (in demselben Änderungslauf) ein Zyklus ist, oder ob diese mehrmalige Änderung infolge der gleichzeitigen Änderung eines anderen identifizierenden Attributs erforderlich ist.
Es ist daher zu prüfen, ob zwei unterschiedliche Klassen von Tabellen für ein relationales Datenbanksystem aufgebaut werden können. Wobei die erste Klasse von Tabellen im Änderungsdienst gepflegt werden würde und daher den Anforderungen der optimalen 3NF genügen muß und die zusätzliche Anforderung keine Zyklen enthalten darf;

und eine zweite Klasse von Tabellen, die solche funktionalen Abhängigkeiten enthält, die zu Zyklen führen. Darüberhinaus könnte man über eine solche zweite Klasse von Tabellen eine eventuell mangelnde Erreichbarkeit beheben und für sehr häufige Abfragen eventuell kürzere Zugriffszeiten erreichen. Eine solche Vorgehensweise wird bei einigen Datenbanksystemen z.B. bei SESAM bereits praktiziert.
Entschließt man nicht zu dieser o.g. Lösung mit zwei unterschiedlichen Klassen von Tabellen, sondern dazu nur eine Art von Tabellen zuzulassen und Zyklen in die Lösung aufzunehmen, so kann man im allgemeinen die - bei der Zyklenauflösung - zunächst eliminierten Kanten in die Lösung wieder aufnehmen.
Bei den bisherigen Beispielen wurde festgestellt, daß dadurch keine transitiven Abhängigkeiten entstehen, wenn die Erreichbarkeit der Knoten untereinander gegeben war. Ist diese Erreichbarkeit (des zyklenfreien Graphen) dagegen nicht gegeben oder sind die Ergebnisse nicht in optimaler 3NF, so können diese zunächst eliminierten Kanten nicht in die Lösung aufgenommen werden ohne daß transitive Abhängigkeiten entstehen (vgl. Anlage 59, 60, 65, 66).

3.6 ALLGEMEINGÜLTIGE MODELLVALIDIERUNG

3.6.1 ÜBERPRÜFUNG DER OPTIMALEN 3NF

In der 3NF sind transitive Abhängigkeiten der Nichtschlüsselattribute (NSA) von den Schlüsselattributen (SA) innerhalb einer Relation unzulässig.[115] In der optimalen 3NF dürfen keine redundanten funktionalen Abhängigkeiten abgebildet werden. Geht man nun von einer graphischen Darstellung der voll funktionalen Abhängigkeiten aus, so lautet die erste Forderung demnach:

> In einem gerichteten Graphen dürfen keine transitiven Zusammensetzungen enthalten sein.

Es wird zunächst davon ausgegangen, daß dieser gerichtete Graph keine Zyklen enthält.
Damit nun die maximale Anzahl voll funktionaler Abhängigkeiten berücksichtigt werden kann lautet die zweite Forderung:

> Es sind nur solche voll funktionalen Abhängigkeiten zu eliminieren, die - infolge transitiver Zusammensetzungen - redundant sind.

Um transitive Zusammensetzungen eines gerichteten Graphen zu eliminieren, muß in jeder Masche des Graphen mindestens eine Kante gestrichen werden.
Um die maximale Anzahl voll funktionaler Abhängigkeiten zu erhalten, darf jedoch höchstens eine Kante in jeder Masche gestrichen werden.
Diese Forderungen erfüllt ein gerichteter Baum gemäß nachfolgender Definitionen:

---

115) vgl. Schlageter, G., Stucky, W.: Datenbanksysteme, a.a.O., S.150 und Kapitel 1.1.6

Definition: Ein zusammenhängender Graph ohne Kreise heißt Baum. [116]

Ein gerichteter Graph X ist eine Menge von Knoten K=K(X) und eine Menge von Kanten E=E(X) mit geordneten Paaren (i,j) verschiedener Elemente aus K. [117]

Ein Digraph D=(K,E) heißt gerichteter Baum mit der Wurzel $r \in K$, wenn der D zugeordnete Graph ein Baum und $\{r\}$ Basis von D ist. [118]

Satz: [119] Es sei X ein Graph mit n Knoten und m Kanten. dann sind folgende Aussagen äquivalent:

a) X ist ein Baum.
b) Je zwei Knoten von X sind durch genau einen Weg in X verbunden.
c) X ist zusammenhängend und jede Kante von X ist eine Brücke.
d) X ist zusammenhängend und m = n-1
e) X besitzt keinen Kreis und es ist m = n-1
f) X besitzt keinen Kreis, verbindet man aber zwei Knoten von K(X), die in X nicht verbunden sind, durch eine Kante, so erhält man einen Graphen mit genau einem Kreis.

Da in einem gerichteten Baum je zwei Knoten i,j von X durch genau einen Pfeil in X verbunden sind (vgl. b) kann dieser gerichtete Graph keine transitiven Zusammensetzungen enthalten.
Ein gerichteter Baum enthält keine Masche (vgl. e). Verbindet man jedoch zwei Knoten i,j, die nicht verbunden sind durch einen Pfeil, so erhält man einen Graphen mit genau einer Masche. Somit ist gewährleistet, daß ein Baum die Forderungen der optimalen 3NF erfüllt.

---

116) vgl. Dörfler, W., Mühlbacher, J.: Graphentheorie für Informatiker, a.a.O., S. 21
117) vgl. ebenda, S. 39
118) vgl. Neumann, K.: Operations Research Verfahren, Bd. 3, a.a.O., S.40
119) vgl. Dörfler, W., Mühlbacher, J.: Graphentheorie für Informatiker, a.a.O., S.23

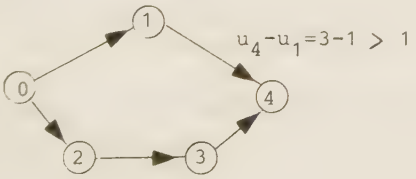
3.6.2 Überprüfung der maximalen Erreichbarkeit der Knoten untereinander

3.6.2.1 Bei gerichteten Graphen ohne Zyklen

Durch einen gerichteten Baum wird zunächst nur die Erreichbarkeit zwischen der Wurzel und allen übrigen Knoten gewährleistet.
Es besteht nun die Forderung die größtmögliche Erreichbarkeit der Knoten untereinander zu erhalten.
Sie wird minimal gestört, wenn:

a) höchstens eine Kante in jeder Masche gestrichen wird,
b) genau die Kante (i,j) gestrichen wird, für die die Menge der Vorgängerknoten $h \in V(j)$ [120] minimal ist,
c) alle Nachfolgerknoten $k \in N(j)$ [121] auch nach dem Streichen von (i,j) erreichbar bleiben, also j durch das Streichen von (i,j) nicht zur Wurzel wird.

Durch die Forderung c) liegt die zu streichende Kante (i,j) stets am Ende einer Masche. Die Forderung b) beinhaltet, daß die letzte Kante des minimalen Weges der Masche zu streichen ist. Sie hat die Eigenschaft $u_j - u_i - 1 > 0$ bzw. $u_j - u_i > 1$ [122]:
Beispiel:



---

120) V(j) ist die Menge der Vorgängerknoten eines Knoten j
121) N(j) ist die Menge der Nachfolgerknoten eines Knoten j
122) u_i ist die maximale Weglänge von der Wurzel des Graphen zum Knoten i.

Enthält eine Masche zwei gleichlange Wege, dann kann wahlweise die letzte Kante einer der beiden Wege gestrichen werden.
Für beide Kanten gilt dann $u_j - u_i = 1$.
Beispiel:

Tritt dieser Fall auf, so existiert mehr als eine Lösung in optimaler 3NF:

Die Forderung genau die Kante (i,j) zu streichen, die am Ende der Masche liegt und für die die Menge der Vorgängerknoten V(j) minimal ist, ist äquivalent zu der Suche nach dem maximalen Weg zwischen der Wurzel und den übrigen Knoten des Graphen.
Werden nun dem Baumalgorithmus - anstelle der allgemein üblichen minimalen Entfernungen - die maximalen Entfernungen zwischen den Knoten zugrundegelegt, wie dies auf S. 131 begründet wurde, so sind die oben genannten Forderungen erfüllt. Wird der Algorithmus von d'Esopo zur Bestimmung der kürzesten Entfernungen zwischen festen Knoten, so modifiziert, daß die maximalen Entfernungen und der zugehörige Baum ermittelt wird, so kann damit die optimale 3NF für gerichtete Graphen ohne Zyklen bestimmt werden.

3.6.2.2 Bei gerichteten Graphen mit Zyklen

Enthält ein Graph Zyklen, dann folgt aus der Bedingung b) (vgl. S. 157), daß die Erreichbarkeit der Knoten untereinander maximiert wird, wenn man in jedem Zyklus die Kante streicht, deren Elimination die Anzahl der Knoten auf dem kürzesten Weg in jeder Masche minimiert bzw. auf dem längsten Weg maximiert. Dies läßt sich mit Hilfe eines LP-Ansatzes realisieren.

3.6.2.2.1 LP-Ansatz für das erste Beispiel [123]

In den nachfolgenden LP-Ansätzen [124] ist
u_i die maximale Länge von der Wurzel zu dem Knoten i
x_{ij} die Kante von Knoten i nach j und
M eine genügend große Zahl
E eine genügend große Obergrenze für den Endknoten

$$u_5 \geq 1$$
$$u_2 \geq 1$$
$$u_2 - u_3 \geq 1 - M(1 - x_{32})$$
$$u_3 - u_2 \geq 1 - M(1 - x_{23})$$
$$u_3 \geq 1$$
$$u_4 \geq 1$$
$$u_4 - u_2 \geq 1$$
$$u_4 - u_3 \geq 1$$

Die Rechnungen werden vereinfacht, falls man das Netz derart ergänzt, daß es nur einen einzigen Anfangs- und Endknoten hat.

$$u_5 \leq E$$
$$u_4 \leq E \qquad E \leq 7$$

Durch die nachfolgende Bedingung wird der Zyklus aufgelöst:

$$x_{23} + x_{32} = 1$$

$$\sum_{i=2}^{5} u_i \rightarrow \max$$

$$x_{ij} = \begin{cases} 0 \\ 1 \end{cases} \quad \text{ganzzahlig}$$

123) vgl. Wedekind, H.: Datenbanksysteme I, S.75 und 3.4.1
124) vgl. Meyer, M., Hansen, K., Rohde, M.: Mathematische Planungsverfahren I, a.a.O., S.152 f

3.6.2.2.2 LP-Ansatz für das zweite Beispiel [125]

$$
\begin{aligned}
u_2 &\geq 1 \\
u_5 &\geq 1 \\
u_5 - u_4 &\geq 1 - M(1 - x_{45}) \\
u_4 &\geq 1 \\
u_4 - u_3 &\geq 1 - M(1 - x_{34}) \\
u_3 - u_2 &\geq 1 \\
u_3 - u_5 &\geq 1 - M(1 - x_{53}) \\
u_6 - u_2 &\geq 1 \\
u_6 - u_4 &\geq 1 \\
u_7 - u_6 &\geq 1 \\
u_7 - u_3 &\geq 1
\end{aligned}
$$

$$u_7 \leq E$$

$$x_{45} + x_{53} + x_{34} = 2$$

$$x_{ij} = \begin{cases} 0 \\ \\ 1 \end{cases} \quad \text{ganzzahlig}$$

$$\sum_{i=2}^{7} u_i \longrightarrow \max$$

125) vgl. Hu, Te Chiang: Ganzzahlige Programmierung, a.a.O., S.180 und 3.4.2

3.6.2.2.3 LP-Ansatz für das dritte Beispiel [126]

$$
\begin{aligned}
u_2 &\geq 1 \\
u_2 - u_4 &\geq 1 - M(1 - x_{42}) \\
u_2 - u_9 &\geq 1 \\
u_3 - u_5 &\geq 1 \\
u_3 - u_2 &\geq 1 - M(1 - x_{23}) \\
u_3 - u_6 &\geq 1 \\
u_4 - u_3 &\geq 1 - M(1 - x_{34}) \\
u_4 - u_7 &\geq 1 \\
u_4 - u_8 &\geq 1 \\
u_5 &\geq 1 \\
u_6 - u_5 &\geq 1 \\
u_7 - u_6 &\geq 1 \\
u_8 - u_7 &\geq 1 \\
u_9 - u_8 &\geq 1
\end{aligned}
$$

$$u_4 \leq E$$

$$x_{42} + x_{23} + x_{34} = 2$$

$$x_{ij} = \begin{cases} 0 \\ \\ 1 \end{cases} \quad \text{ganzzahlig}$$

$$\sum_{i=2}^{9} u_i \longrightarrow \max$$

126) vgl. 3.4.3

3.6.2.2.4 LP-Ansatz für das vierte Beispiel [127]

$$
\begin{aligned}
u_2 &\geq 1 - M(1 - x_{21}) \\
u_2 - u_3 &\geq 1 - M(1 - x_{32}) \\
u_3 - u_2 &\geq 1 - M(1 - x_{23}) \\
u_4 - u_3 &\geq 1 \\
u_4 &\geq 1 \\
u_4 - u_5 &\geq 1 - M(1 - x_{54}) \\
u_5 &\geq 1 \\
u_5 - u_4 &\geq 1 - M(1 - x_{45}) \\
u_6 - u_5 &\geq 1 \\
u_6 - u_4 &\geq 1
\end{aligned}
$$

$$u_6 \leq E$$

$$x_{21} + x_{32} + x_{23} = 2$$

$$x_{45} + x_{54} = 1$$

$$x_{ij} = \begin{cases} 0 \\ \\ 1 \end{cases} \quad \text{ganzzahlig}$$

$$\sum_{i=2}^{6} u_i \rightarrow \max$$

127) vgl. Neumann, K.: Operations Research Verfahren Bd. III, a.a.O., S.89 und 3.4.4

3.6.2.2.5 LP-Ansatz für das fünfte Beispiel [128]

$$
\begin{aligned}
u_2 &\geq 1 \\
u_3 - u_2 &\geq 1 \\
u_3 - u_4 &\geq 1 - M(1 - x_{43}) \\
u_4 - u_2 &\geq 1 \\
u_4 - u_3 &\geq 1 - M(1 - x_{34}) \\
u_5 - u_2 &\geq 1 \\
u_5 - u_3 &\geq 1 \\
u_5 - u_4 &\geq 1 \\
u_6 - u_2 &\geq 1 \\
u_7 &\geq 1 \\
u_7 - u_9 &\geq 1 - M(1 - x_{97}) \\
u_8 &\geq 1 \\
u_9 - u_7 &\geq 1 - M(1 - x_{79})
\end{aligned}
$$

$$
\begin{aligned}
u_5 &\leq E \\
u_6 &\leq E \\
u_8 &\leq E \\
u_9 &\leq E
\end{aligned}
$$

$$
\begin{aligned}
x_{34} + x_{43} &= 1 \\
x_{97} + x_{79} &= 1
\end{aligned}
$$

$$
x_{ij} = \begin{cases} 0 \\ \\ 1 \end{cases} \quad \text{ganzzahlig}
$$

$$
\sum_{i=2}^{9} u_i \rightarrow \max
$$

128) vgl. Vetter, M.: Data Base Design by Appplied Data Synthesis, a.a.O., S.431 und 3.4.5

3.6.3 VERFAHRENSVERGLEICH

Um Datenmaterial für ein relationales Datenmodell zu strukturieren müssen

- geeignete Relationen gefunden werden
- diese Relationen normalisiert werden
- und redundante Relationen eliminiert werden

Deshalb wurden im Kapitel 2.1 Verfahren untersucht, die auf einer Datenverwendungsmatrix aufbauen und so aufgrund bekannter Anforderungen - durch Clusteranalyse - Relationen bilden (vgl. Anhang 73). Im Kapitel 2.2 wurde ein Verfahren untersucht, das durch Generalisierung und Typisierung Relationen findet. Dabei handelt es sich jedoch um eine generelle Vorgehensweise bei der Analyse des Datenmaterials, die geeignet ist voll funktionale Abhängigkeiten zu definieren, nicht aber um eine rechentechnisch unterstützte Methode. Eine dritte Gruppe von Verfahren, sie wird im Kapitel 2.3 beschrieben, findet durch Bestimmung der funktionalen Abhängigkeiten Relationen und normalisiert sie.
Da die 3NF von voll funktionalen Abhängigkeiten ausgeht, sind die im Kapitel 2.1 beschriebenen Verfahren weniger geeignet. Sie bilden die Relationen auf der Basis bereits bekannter und ähnlicher Anforderungen, somit besteht kein Zwang, daß alle Attribute einer Relation vom Schlüssel abhängig sind. Darüberhinaus ist der Rechenaufwand verhältnismäßig hoch und die Ergebnisse stark vom zu definierenden Ähnlichkeitsmaß abhängig. Es erscheint daher günstiger, wie bei den Verfahren der Relationensynthese von EFR auszugehen.
Zur Bestimmung der EFR kann das Verfahren von Smith und Smith (vgl. Kapitel 2.2) hilfreich sein.

Eine Berücksichtigung der Zugriffsart (vgl. Anhang 73), wie sie bei Döringer vorgenommen und im Kapitel 2.1.5 beschrieben wird, scheint im ersten Moment sinnvoll. Eine Datenbank wird jedoch meist nur zum Teil für bereits bekannte Anforderungen konzipiert. Ein nicht unerheblicher Teil der Anforderungen werden ad hoc gestellt und sind zum Zeitpunkt der Konzeption nur schwer vorhersehbar. Daher ist es günstiger die Bedeutung der Attribute durch ihre funktionalen Abhängigkeiten zu bestimmen und nicht durch die Zugriffsart oder durch die Reihenfolge der Ordnungskriterien (obwohl diese implizit in den funktionalen Abhängigkeiten enthalten ist). Auch die Berücksichtigung der Zugriffshäufigkeit ist nur sinnvoll bei sonst gleichwertigen Lösungen, zumal auch die Zugriffshäufigkeit schlecht vorhersehbar ist.

Bei der Normalisierung sollte nicht davon ausgegangen werden, daß eine funktionale Abhängigkeit durch genau zwei andere funktionale Abhängigkeiten ersetzt werden kann. Das würde voraussetzen, daß vor der Normalisierung eine Vollständigkeitsprüfung der EFR vorgenommen werden muß. Selbst dann kann es jedoch - wie in dem auf S.112 beschriebenen Fall - vorkommen, daß durch das Eliminieren einer transitiven Abhängigkeit (im Beispiel 1 → 5), transitive Abhängigkeiten über je zwei funktionale Abhängigkeiten bleiben (im Beispiel: r7 + r10 und r8 + r11). Diese transitiven Abhängigkeiten werden bei o.g. Einschränkung nicht weiter bearbeitet. Das neu entwickelte Verfahren unterliegt keinen Einschränkungen bei den transitiven Zusammensetzungen der funktionalen Abhängigkeiten (Ausnahme: der Graph darf nicht mehrere Wurzel haben, da nur die Erreichbarkeit einer Wurzel gewährleistet wird).
Weitere Probleme tauchen immer dann auf, wenn der Graph aus funktionalen Abhängigkeiten Zyklen aufweist.[128)]

---

128) vgl. Härder, Th.: Zur Synthese von Relationen in optimaler dritter Normalform, a.a.O., S. 2 ff
und Hsu, H.T.: An Algorithm for Finding a Minimal Equivalent Graph of a Digraph, a.a.O., S. 11 ff

Bei Härder wird das Problem dadurch gelöst, daß jeweils ein Zyklus zunächst zu einem Hilfsknoten zusammengefaßt wird, Hsu ermittelt vorab eine minimale Überdeckung für den Graphen ohne Zyklen. In dem neu entwickelten Verfahren können Zyklen - auch sich überlagernde Zyklen - durch Lineare Programmierung so aufgelöst werden, daß die maximale Erreichbarkeit der Attribute erhalten wird.

Die Normalisierung in optimale 3NF wird bei dem Verfahren von Döringer (vgl. Kapitel 2.1.5) zwar angestrebt, aber nicht garantiert. Da es sich auch nicht um ein rechentechnisch unterstütztes Verfahren handelt, wurde es für die weitere Betrachtung nicht herangezogen.
In den, im Kapitel 2.3 beschriebenen Verfahren, wird die optimale 3NF (bzw. BCNF oder 4NF) erreicht. Diese Verfahren werden bei der Zyklenbehandlung - sofern sie sie vorsehen - sehr rechenaufwendig. Auch wurde gezeigt, daß - möglicherweise durch Bedienungsfehler - die Ergebnisse auch Relationen aufweisen können, die nicht in optimaler 3NF sind. Ein Datenbankadministrator, der ein solches Verfahren einsetzt, muß sich jedoch darauf verlassen können, daß die Ergebnisse mindestens der optimalen 3NF entsprechen oder er muß gegebenenfall eine entsprechende Fehlermeldung erhalten. Diese ist in den beschriebenen Verfahren nicht vorgesehen.
In dem neu entwickelten Verfahren wird der Begriff der Redundanzfreiheit soweit ausgedehnt, daß ein Attribut durch genau eine funktionale Abhängigkeit bestimmt wird. Dadurch wird der Aufwand im Änderungsdienst noch weiter verringert. Das Ergebnis kann dann keine Zyklen (auch keine inversen Abhängigkeiten) enthalten. Dies ist insofern gerechtfertigt, als ein Zyklus aus voll funktionalen Abhängigkeiten letztlich immer auf eine volle funktionale Abhängigkeit eines Attributs von sich selbst zurückgeführt

werden kann. Diese Bedingung hat jedoch keinen Informationsgehalt und kann ebenfalls zu Anormalitäten im Änderungsdienst führen. Anormalitäten im Änderungsdienst sollen jedoch gerade durch die optimale 3NF vermieden werden.
Durch das Verfahren von d'Esopo zur Bestimmung der Weglängen wird eine rechenzeitgünstige Lösung gewährleistet. Dadurch daß die Bestimmung der Distanzen zwischen den Knoten und die Ermittlung des Baumes in demselben Rechenschritt erfolgt, ist das Verfahren mit Sicherheit schneller als die Verfahren von Wang,Wedekind und Vetter, bei denen der minimalen Überdeckung die Bestimmung der Distanzen vorausgeht.
Prinzipiell ist das neue Verfahren geeignet um alle Lösungsmöglichkeiten zu finden. Im Programm wird jedoch nur eine zulässige Lösung ermittelt. Bei gleichlangen Wegen in einer Masche gibt es mehrere Lösungsmöglichkeiten. In diesem Fall entscheidet die Zugriffshäufigkeit welcher Weg gewählt wird. Insoweit ist die Berücksichtigung der Häufigkeit des Datenzugriffs durch die Bewertung der Knoten des Graphen möglich.
Durch den enger gefaßten Begriff der Redundanzfreiheit besteht bei dem neuen Verfahren die Gefahr, daß die Erreichbarkeit der Attribute verlorengeht. Aus diesem Grund wird die Erreichbarkeit - die ursprünglich durch die funktionalen Abhängigkeiten gegeben war - überprüft und gegebenenfalls eine Fehlermeldung gedruckt. Lediglich bei dem Verfahren von Vetter (vgl. Anhang 73) wird die Erreichbarkeit der Attribute aufgrund bereits bekannter Anforderungen überpüft. Da aber - wie bereits erwähnt - in der Konzeptionsphase nicht alle Anforderungen bekannt sein können, ist es besser die Erreichbarkeit aufgrund der funktionalen Abhängigkeiten zu prüfen.

4. SCHLUSSBETRACHTUNG

Für die im zweiten Kapitel untersuchten Clusteranalyseverfahren wurde ein Testprogramm in PL/I geschrieben. Die Beispiele der Clusteranalyse wurden auf einer Siemens-Anlage 7-531 gerechnet.
Für die Beispiele der Relationensynthese wurden Testprogramme in FORTRAN IV geschrieben. Diese Beispiele wurden auf einer CDC-Anlage CYBER 175 gerechnet.
Mit diesen Testprogrammen sollte primär das Verhalten der Verfahren an diversen Beispielen getestet werden.
Die Verfahren wurden so implementiert, wie sie in der Literatur beschrieben waren. Dabei wurde Wert auf möglichst einfache und übersichtliche Programmierung gelegt, weshalb die Ablaufgeschwindigkeit der Testprogramme als zweitrangig betrachtet wurde. Da die Testprogramme somit nicht ablaufoptimal geschrieben waren, wurde kein Zeitvergleich vorgenommen.

Die Aussage, daß das neu entwickelte Verfahren schneller als die anderen Verfahren ist, stützt sich auf folgende Punkte:

- Bei dem Verfahren von Wedekind ist Voraussetzung, daß die maximalen Distanzen zwischen den Knoten vor Ablauf des eigentlichen Verfahrens bekannt sind. [129)] Unabhängig von der Ablaufgeschwindigkeit des eigentlichen Verfahrens von Wedekind, muß man feststellen, daß der Algorithmus zur Bestimmung der maximalen Distanzen nach d'Esopo den zugehörigen Baum lediglich durch zusätzliche Übertragungsbefehle liefert, die bekanntermaßen kaum CPU-Zeit benötigen.
- In einem Vergleich von Pape [130)], dem exakte Zeitmessungen

129) vgl. Wedekind, H.: Datenbanksysteme I, a.a.O.,S.75 ff
130) vgl. Pape, U.: Kürzeste Wege in Netzwerken zwischen festen Knoten, a.a.O., S. 52

zugrundeliegen, wurde das Verfahren von d'Esopo als das schnellste Vefahren zur Bestimmung der Distanzen zwischen festen Knoten erkannt. Es benötigt ca. 40% der CPU-Zeit, die nach dem Verfahren von Dijkstra benötigt wird und ca. 60% der CPU-Zeit, die nach dem Verfahren von Moore benötigt wird.

- Bei dem Verfahren von Vetter [131] werden alle Rechenoperationen sowohl mit den 1-Elementen als auch mit den 0-Elementen der Adjazenzmatrix durchgeführt. Dies ist aufwendiger als bei dem Verfahren von d'Esopo, bei dem nur die 1-Elemente berücksichtigt werden.

Die Aussage, daß das neu entwickelte Verfahren weniger Speicherplatz benötigt, als die anderen Verfahren, beruht auf folgenden Fakten:

- Das Verfahren von d'Esopo geht von der Listendarstellung des Graphen aus. [132] Daher ist der Speicherplatzbedarf umso geringer, je weniger Elemente der Adjazenzmatrix den Wert 1 haben.
- Der Algorithmus von Vetter beruht auf einer Speicherung der Daten in Matrixform. Damit ist der Speicherplatzbedarf bei dünnbesetzten Matrizen relativ groß.
- Bei dem Verfahren von Wedekind, werden bei jeder Iteration umfangreiche Tabellen gespeichert. Insoweit benötigt dieses Verfahren wesentlich mehr Speicherplatz als das Verfahren von d'Esopo.

Die Sicherheit, daß die erzielten Ergebnisse immer den gewünschten Anforderungen genügen ist folgendermaßen begründet:

- Ein Baum - als Ergebnis - kann per Definiton keine transitiven Abhängigkeiten enthalten.

---

131) vgl. Vetter, M.: Data Base Design by Applied Data Synthesis, a.a.O., S. 428 ff

132) vgl. Dörfler, W., Mühlbacher, J.: Graphentheorie für Informatiker, a.a.O., S.63f

- Ein Baum, dem die maximalen Entfernungen zwischen den Knoten zugrundeliegen, gewährleistet die maximale Erreichbarkeit der Knoten untereinander. Daß dies immer zutreffen muß wurde ausführlich im Kapitel 3.6.1 und 3.6.2 gezeigt.
- Die Erreichbarkeit der Knoten untereinander wird geprüft. Dadurch wird vermieden, daß das Verfahren durch falsche Anwendung oder fehlerhafte Implementierung zu fehlerhaften Ergebnissen führt.

Derartige fehlerhafte Ergebnisse konnten beispielsweise mit dem Verfahren von Vetter durch die Angabe unzweckmäßiger Gewichtungen erreicht werden (vgl. S. 100 und Anhang 45). Ebenso wurde mit einem Beispiel bei dem Verfahren von Wedekind ein Ergebnis erzielt, das nicht in optimaler 3NF ist (vgl. S.112 und Anhang 51). Dieses Ergebnis ist möglicherweise auf die Implementierung zurückzuführen, da dieser spezielle Fall bei Wedekind nicht explizit beschrieben wurde. In den untersuchten Verfahren wäre deshalb eine Überprüfung der Ergebnisse wünschenswert, die dem Anwender die Sicherheit geben würde, daß die Ergebnisse ohne Fehlermeldung in optimaler 3NF sind und bei fehlerhafter Eingabe oder fehlerhafter Anwendung des Verfahrens das Ergebnis entsprechend kommentiert würde.

Das Testprogramm für das neue Verfahren könnte in folgenden Punkten optimiert werden:

- Die Ein- und Ausgabe der Daten wurde in Matrixform vorgenommen. Einerseits dient dies der besseren Anschaulichkeit andererseits wurde durch einheitliche Ein-/Ausgaberoutinen der Aufwand für die Testprogramme insgesamt verringert.
  Der Vorteil des neuen Verfahrens liegt nicht zuletzt in der günstigen Speicherplatzverwaltung des Verfahrens von d'Esopo. Soll die Ablaufgeschwindigkeit des Programms optimiert werden, so ist ein Wechsel der Speicherungsform des Graphen von Matrixdarstellung zur Listendarstellung zu vermeiden.

- Da in diesem Verfahren große Datenmengen bearbeitet werden sollen, wäre eine dynamische Speicherplatzverwaltung empfehlenswert. Bei einer solchen Lösung müßte die Anzahl der zu verarbeitenden Attribute nicht per Programm begrenzt werden, sondern wäre nur durch den zur Verfügung stehenden Kernspeicher begrenzt.
- Eine weitere Verbesserung der Erreichbarkeit könnte möglicherweise dadurch erreicht werden, daß alle implizit gegebenen funktionalen Abhängigkeiten maschinell ermittelt und zusätzlich zu den explizit angegebenen funktionalen Abhängigkeiten bearbeitet würden.
- Zur Überprüfung der Erreichbarkeit der Knoten untereinander wurde ein Verfahren gesucht, das anders vorgeht, als das bereits angewendete Verfahren zur Distanzermittlung.Deshalb wurde die Erreichbarkeit der Knoten durch Potenzieren der Adjazenzmatrix ermittelt. [133] Hier könnte zu Prüfzwecken ein schnelleres Verfahren eingesetzt werden, das dennoch anders als das Verfahren von d'Esopo arbeitet.

Eine Verbesserung des eigentlichen Verfahrens wäre durch eine Änderung der wesentlichen Einschränkungen des Verfahrens gegeben:

- Es wäre zu prüfen ob Attributgruppen als Schlüssel zugelassen werden können. Inwieweit dies durch Erweiterung des neuen Verfahrens oder durch ein vorgelagertes Verfahren erreicht werden kann, wäre detailliert zu untersuchen.
- Die Einschränkung des neuen Verfahrens auf Graphen mit nur einer Wurzel resultiert aus der Definition des Baumes.

---

133) vgl. Neumann, K.: Operations Research Verfahren Bd. 3, a.a.O., S. 48 ff

Würden in diesem Verfahren mehrere Wurzeln zugelassen, so würde im Ergebnis die Erreichbarkeit einzelner Attribute (von diesen zusätzlichen Wurzeln aus) verlorengehen. Es müßte ohne weiteres möglich sein, diese Erreichbarkeit zu berücksichtigen und trotzdem sicherzustellen, daß die Ergebnisse in optimaler 3NF sind. In diesem Fall müßte jedoch von der Definiton des Ergebnisses als Baum abgegangen werden.

- Ebenso wäre eine Erweiterung des Verfahrens auf die Belange der 4NF denkbar. Voraussetzung wäre eine Kennzeichnung der multivalued dependencies für das Verfahren. Möglicherweise müßte dann eine Sonderbehandlung dieser multivalued dependencies durchgeführt werden.

VERZEICHNIS DES ANHANGS

zu Kapitel 2.1

Anhang 1 bis 10	Clusteranalyse der Matrix A
Anhang 11 bis 24	Clusteranalyse der Matrix B
Anhang 25 bis 32	Clusteranalyse der Matrix D
Anhang 33 bis 37	Clusteranalyse der Matrix K

zu Kapitel 2.2:

Anhang 38 bis 45	Relationensynthese für das erste Beispiel
Anhang 46 bis 50	Relationensynthese für das zweite Beispiel
Anhang 51 bis 52	Relationensynthese für das dritte Beispiel

zu Kapitel 3.5:

Anhang 53 bis 54	Ergebnisse des ersten Beispiels
Anhang 55 bis 60	Ergebnisse des zweiten Beispiels
Anhang 61 bis 64	Ergebnisse des dritten Beispiels
Anhang 65 bis 68	Ergebnisse des vierten Beispiels
Anhang 69 bis 72	Ergebnisse des fünften Beispiels

zu Kapitel 4:

Anhang 73	Verfahrensübersicht

CLUSTERANALYSE VERFAHREN

MATRIX = INPUT DATENMATRIX

i \ j	1	2	3	4	5	6	7	8	9	10
1	0	0	0	1						
2	1	1	1	0						
3	0	0	1	0						
4	1	1	0	0						
5	0	0	1	0						

CLUSTERANALYSE VERFAHREN

MATRIX = AEHNLICHKEITSMATRIX N.FREUD.

i \ j	1	2	3	4	5	6	7	8	9
1	4	0	2	1	2				
2	0	4	2	3	2				
3	2	2	4	1	4				
4	1	3	1	4	1				
5	2	2	4	1	4				

Anhang 1: Matrix A und Ähnlichkeitsmatrix nach Freudenthaler und Maier

Quelle: Freudenthaler, B., Maier,B.: a.a.O.S.524

CLUSTERANALYSE VERFAHREN

MATRIX = W-MATRIX KLASSIFIKATIONSNIVEAU 1

i \ t	1	2	3	4	5	6	7	8	9
1	1	0	1	1	1				
2	0	1	1	1	1				
3	1	1	1	1	1				
4	1	1	1	1	1				
5	1	1	1	1	1				

CLUSTERANALYSE VERFAHREN

MATRIX = BAUM

	1	2	3	4	5	6	7	8	9
1	3	1	4	5					
2	3	2	4	5					

Anhang 2: *I*-Matrix und Baumbildung auf dem Klassifikationsn
1 (aus Matrix A)

CLUSTERANALYSE VERFAHREN

MATRIX = W-MATRIX KLASSIFIKATIONSNIVEAU 2

i \ t	1	2	3	4	5	6	7	8	9	10
1	1	0	1	0	1					
2	0	1	1	1	1					
3	1	1	1	0	1					
4	0	1	0	1	0					
5	1	1	1	0	1					

CLUSTERANALYSE VERFAHREN

MATRIX = BAUM

	1	2	3	4	5	6	7	8	9	1
1	3	1	5							
2	3	2	5							
3	4	2	0							

Anhang 3: W-Matrix und Baumbildung auf dem Klassifikationsniveau 2 (aus Matrix A)

CLUSTERANALYSE VERFAHREN

MATRIX = W-MATRIX KLASSIFIKATIONSNIVEAU 3

I \ K	1	2	3	4	5	6	7	8	9
1	1	0	0	0	0				
2	0	1	0	1	0				
3	0	0	1	0	1				
4	0	1	0	1	0				
5	0	0	1	0	1				

CLUSTERANALYSE VERFAHREN

MATRIX = BAUM

	1	2	3	4	5	6	7	8	9
1	2	4							
2	1	0							
3	3	5							

Anhang 4: W-Matrix und Baumbildung auf dem Klassifikationsniveau 3 (aus Matrix A)

CLUSTERANALYSE VERFAHREN

MATRIX = W-MATRIX KLASSIFIKATIONSNIVEAU 4

i \ k	1	2	3	4	5	6	7	8	9
1	1	0	0	0	0				
2	0	1	0	0	0				
3	0	0	1	0	1				
4	0	0	0	1	0				
5	0	0	1	0	1				

CLUSTERANALYSE VERFAHREN

MATRIX = BAUM

	1	2	3	4	5	6	7	8	9
1	3	5							
2	1	0							
3	2	0							
4	4	0							

Anhang 5: W-Matrix und Baumbildung auf dem Klassifikationsniveau 4 (aus Matrix A)

CLUSTERANALYSE VERFAHREN

MATRIX = W-MATRIX KLASSIFIKATIONSNIVEAU 1

I \ J	1	2	3	4	5	6	7	8	9
1	1	1	1	1					
2	1	1	1	1					
3	1	1	1	1					
4	1	1	1	1					

CLUSTERANALYSE VERFAHREN

MATRIX = BAUM

	1	2	3	4	5	6	7	8	9
1	1	2	3	4					

Anhang 5: W-Matrix und Baumbildung auf dem Klassifikationsniveau 1 (aus transponierter Matrix A)

STERANALYSE VERFAHREN

RIX = W-MATRIX KLASSIFIKATIONSNIVEAU 2

i \ k	1	2	3	4	5	6	7	8	9	10
	1	1	1	1						
	1	1	1	1						
	1	1	1	0						
	1	1	0	1						

USTERANALYSE VERFAHREN

TRIX = BAUM

	1	2	3	4	5	6	7	8	9	10
1	1	2	3							
2	1	3	0							
3	1	4	0							

Anhang 7: W-Matrix und Baumbildung auf dem Klassifikationsniveau 2 (aus transponierter Matrix A)

CLUSTERANALYSE VERFAHREN

MATRIX = W-MATRIX KLASSIFIKATIONSNIVEAU 3

i \ j	1	2	3	4	5	6	7	8	9
1	1	1	0	0					
2	1	1	0	0					
3	0	0	1	0					
4	0	0	0	1					

CLUSTERANALYSE VERFAHREN

MATRIX = BAUM

	1	2	3	4	5	6	7	8	9
1	1	2							
2	3	0							
3	4	0							

Anhang 8: W-Matrix und Baumbildung auf dem Klassifikationsniveau 3 (aus transponierter Matrix A)

...TERANALYSE VERFAHREN

TRIX = W-MATRIX KLASSIFIKATIONSNIVEAU 4

	1	2	3	4	5	6	7	8	9	10
	1	1	0	0						
	1	1	0	0						
	0	0	1	0						
	0	0	0	1						

LUSTERANALYSE VERFAHREN

ATRIX = BAUM

	1	2	3	4	5	6	7	8	9	10
1	1	2								
2	3	0								
3	4	0								

Anhang 9: W-Matrix und Baumbildung auf dem Klassifikationsniveau 4 (aus transponierter Matrix A)

CLUSTERANALYSE VERFAHREN

MATRIX = W-MATRIX KLASSIFIKATIONSNIVEAU 5

i N	1	2	3	4	5	6	7	8	9
1	1	1	0	0					
2	1	1	0	0					
3	0	0	1	0					
4	0	0	0	1					

CLUSTERANALYSE VERFAHREN

MATRIX = BAUM

	1	2	3	4	5	6	7	8	9
1	1	2							
2	3	0							
3	4	0							

Anhang 10: W-Matrix und Baumbildung auf dem Klassifikationsniveau 5 (aus transponierter Matrix A)

MATRIX = INPUT DATENMATRIX

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	1	1	1	0	0	0	1	0	0	0	1	0	0	0	0	0	1	0
2	1	1	1	0	0	0	1	0	0	0	1	0	0	0	0	0	1	0
3	1	1	1	0	0	0	1	0	0	0	1	0	0	0	0	0	1	0
4	1	1	1	0	0	0	1	0	0	0	1	0	0	0	0	0	1	0
5	1	1	1	0	0	0	1	0	0	0	1	0	0	0	0	0	1	0
6	1	1	1	0	0	0	1	0	0	0	1	0	0	0	0	0	1	0
7	1	1	1	0	0	0	1	0	0	0	1	0	0	0	0	0	1	0
8	1	1	1	0	0	0	1	0	0	0	1	0	0	0	0	0	1	0
9	1	1	1	0	0	0	1	0	0	0	1	0	0	0	0	0	1	0
10	1	1	1	0	0	0	1	0	0	0	1	0	0	0	0	0	1	0
11	1	1	1	0	0	0	1	0	0	0	1	0	0	0	0	0	1	0
12	1	1	1	0	0	0	1	0	0	0	1	0	0	0	0	0	1	0
13	1	1	1	0	0	0	1	0	0	0	1	0	0	0	0	0	1	0
14	1	1	1	0	0	0	1	0	0	0	1	0	0	0	0	0	1	0
15	1	1	1	0	0	0	1	0	0	0	1	0	0	0	0	0	1	0
16	1	1	1	0	0	0	1	0	0	0	1	0	0	0	0	0	1	0
17	1	1	1	0	0	0	1	0	0	0	1	0	0	0	0	0	1	0
18	1	1	1	0	0	0	1	0	0	0	1	0	0	0	0	0	1	0
19	1	1	1	0	0	0	1	0	0	0	1	0	0	0	0	0	1	0
20	1	1	1	0	0	0	1	0	0	0	1	0	0	0	0	0	1	0
21	1	1	1	0	0	0	1	0	0	0	1	0	0	0	0	0	1	0
22	1	1	1	0	0	0	1	0	0	0	1	0	0	0	0	0	1	0
23	1	1	1	0	0	0	1	0	0	0	1	0	0	0	0	0	1	0
24	1	1	1	0	0	0	1	0	0	0	1	0	0	0	0	0	1	0
25	1	1	1	0	0	0	1	0	0	0	1	0	0	0	0	0	1	0
26	1	1	1	0	0	0	1	0	0	0	1	0	0	0	0	0	1	0
27	1	1	1	0	0	0	1	0	0	0	1	0	0	0	0	0	1	0
28	1	1	1	0	0	0	1	0	0	0	1	0	0	0	0	0	1	0
29	1	1	1	0	0	0	1	0	0	0	1	0	0	0	0	0	1	0
30	1	1	1	0	0	0	1	0	0	0	1	0	0	0	0	0	1	0
31	1	1	1	0	0	0	1	0	0	0	1	0	0	0	0	0	1	0

Anhang 11: Matrix B, vgl. Freudenthaler,B., Maier,B.: a.a.O.,S. 327

Anhang 12: Matrix B, Ähnlichkeitsmatrix nach dem Verfahren von Freudenthaler und Maier

MATRIX = AEHNLICHKEITSMATRIX N.FREUD.

M/N	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
31	7	9	11	14	12	11	11	11	12	15	16	16	16	0	0	0	0	0

Anhang 12, Bl. 2

Anhang 13: W-Matrix auf dem Klassifikationsniveau 13
(zu Matrix B)

CLUSTERANALYSE VERFAHREN DATUM 28.04.80 SEITE 7

MATRIX = W-MATRIX KLASSIFIKATIONSNIVEAU 13

M/N	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
31	0	0	0	1	0	0	0	0	0	1	1	1	1	0	0	0	0	0

Anhang 13: Blatt 2

Anhang 14: Subgraph auf dem Klassifikationsniveau 13
(zu Matrix B)

MATRIX = BAUM

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	22	15	24	25	26	0												
2	22	21	0	0	0	0												
3	22	24	23	26	0	0												
4	22	25	28	0	0	0												
5	22	26	0	0	0	0												
6	22	28	29	30	31	0												
7	1	2	6	7	0	0												
8	3	5	0	0	0	0												
9	3	5	0	0	0	0												
10	3	8	0	0	0	0												
11	9	10	12	13	14	0												
12	11	13	14	0	0	0												
13	11	14	0	0	0	0												
14	11	17	0	0	0	0												
15	16	12	13	14	0	0												
16	16	13	14	0	0	0												
17	16	14	0	0	0	0												
18	16	17	23	0	0	0												
19	18	27	0	0	0	0												
20	19	20	0	0	0	0												
21	23	24	25	26	27	28												

Anhang 15: Baumbidlung auf dem Klassifikationsniveau 13
(Matrix B)

MATRIX = TANIMOTO ABSTANDSMATRIX

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	0.00	0.00	0.62	0.62	0.50	0.16	0.16	0.57	0.63	0.53	0.89	0.69	0.72	0.79	1.00	0.89	1.00	1.00
2	0.00	0.00	0.62	0.62	0.50	0.16	0.16	0.57	0.63	0.53	0.89	0.69	0.72	0.79	1.00	0.89	1.00	1.00
3	0.62	0.62	0.00	0.00	0.37	0.50	0.50	0.42	0.53	0.45	0.66	0.59	0.50	0.55	0.87	0.66	0.77	1.00
4	0.62	0.62	0.00	0.00	0.37	0.50	0.50	0.42	0.53	0.45	0.66	0.59	0.50	0.55	0.87	0.66	0.77	1.00
5	0.50	0.50	0.37	0.37	0.00	0.37	0.37	0.46	0.57	0.50	0.81	0.75	0.63	0.72	0.75	0.81	0.90	0.88
6	0.16	0.16	0.50	0.50	0.37	0.00	0.00	0.62	0.53	0.45	0.79	0.72	0.63	0.69	1.00	0.90	1.00	1.00
7	0.16	0.16	0.50	0.50	0.37	0.00	0.00	0.62	0.53	0.45	0.79	0.72	0.63	0.69	1.00	0.90	1.00	1.00
8	0.57	0.57	0.42	0.42	0.46	0.62	0.62	0.00	0.63	0.55	0.77	0.55	0.59	0.79	1.00	0.82	0.75	1.00
9	0.63	0.63	0.53	0.53	0.57	0.53	0.53	0.63	0.00	0.08	0.39	0.19	0.09	0.29	0.91	0.39	0.50	0.91
10	0.53	0.53	0.45	0.45	0.50	0.45	0.45	0.55	0.08	0.00	0.45	0.26	0.17	0.35	0.92	0.45	0.53	0.92
11	0.89	0.89	0.66	0.66	0.81	0.79	0.79	0.77	0.39	0.45	0.00	0.43	0.32	0.37	0.87	0.50	0.42	1.00
12	0.69	0.69	0.59	0.59	0.75	0.72	0.72	0.55	0.19	0.26	0.43	0.00	0.10	0.32	0.89	0.25	0.37	0.89
13	0.72	0.72	0.50	0.50	0.63	0.63	0.63	0.59	0.09	0.17	0.32	0.10	0.00	0.21	0.90	0.32	0.43	0.90
14	0.79	0.79	0.55	0.55	0.72	0.69	0.69	0.79	0.29	0.35	0.37	0.32	0.21	0.00	0.88	0.37	0.50	0.88
15	1.00	1.00	0.87	0.87	0.75	1.00	1.00	0.91	0.92	0.45	0.89	0.90	0.88	0.88	0.00	0.87	0.85	0.79
16	0.89	0.89	0.66	0.66	0.81	0.90	0.90	0.62	0.39	0.45	0.50	0.26	0.17	0.37	0.87	0.00	0.53	0.87
17	1.00	1.00	0.77	0.77	0.90	0.88	1.00	0.75	0.50	0.92	1.00	0.42	0.37	0.43	0.50	0.88	0.00	0.85
18	1.00	1.00	1.00	1.00	0.88	1.00	1.00	0.88	0.91	0.92	1.00	0.89	0.91	0.92	0.88	0.79	0.85	0.00
19	1.00	1.00	0.84	0.84	0.85	1.00	1.00	0.82	0.64	0.76	0.83	0.57	0.90	0.96	0.96	0.87	0.85	0.85
20	1.00	1.00	0.81	0.81	0.82	1.00	1.00	0.78	0.70	0.81	0.75	0.76	0.72	0.57	0.96	0.69	0.43	0.96
21	1.00	1.00	0.87	0.87	0.88	1.00	1.00	0.85	0.69	0.72	0.77	0.62	0.66	0.75	0.79	0.85	0.50	0.79
22	1.00	1.00	0.85	0.85	0.87	1.00	1.00	0.85	0.69	0.91	0.85	0.88	0.90	0.87	0.79	0.85	0.86	0.82
23	1.00	1.00	1.00	1.00	0.89	1.00	1.00	0.87	0.92	0.92	0.88	0.90	0.91	1.00	0.82	0.88	0.87	0.82
24	1.00	1.00	0.89	0.89	0.79	1.00	1.00	0.92	0.92	0.92	0.89	0.91	0.92	0.90	0.39	0.89	0.88	0.88
25	1.00	1.00	0.89	0.89	0.79	1.00	1.00	0.92	0.92	0.92	0.89	0.91	0.92	0.90	0.39	0.89	0.88	0.88
26	1.00	1.00	0.89	0.89	0.79	1.00	1.00	0.92	0.92	0.89	0.91	0.92	0.90	0.39	0.89	0.88	0.88	0.88
27	1.00	1.00	0.87	0.87	0.85	1.00	1.00	0.89	1.00	1.00	1.00	1.00	1.00	0.59	1.00	1.00	1.00	0.59
28	1.00	1.00	1.00	1.00	0.85	1.00	1.00	0.89	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
29	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
30	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
31	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
1	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
2	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
3	0.84	0.84	0.81	0.81	0.85	1.00	1.00	0.89	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
4	0.84	0.84	0.81	0.81	0.85	1.00	1.00	0.89	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
5	0.85	0.85	0.82	0.82	0.88	0.89	0.89	0.79	0.89	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
6	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
7	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
8	0.82	0.90	0.85	1.00	0.87	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
9	0.84	0.84	0.69	0.90	0.92	0.92	0.92	0.92	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
10	0.66	0.66	0.78	0.78	0.92	0.92	0.92	0.92	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
11	0.63	0.63	0.81	0.71	0.85	0.88	0.89	0.89	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
12	0.57	0.76	0.75	0.62	0.88	0.90	0.91	0.91	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
13	0.60	0.76	0.86	0.68	0.89	0.90	0.91	0.92	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
14	0.66	0.66	0.72	0.75	0.87	1.00	0.90	0.90	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
15	0.66	0.66	0.77	0.77	0.82	0.89	0.89	0.39	0.59	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
16	0.50	0.50	0.69	0.50	0.85	0.88	0.89	0.89	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
17	0.63	0.69	0.59	0.59	0.82	0.88	0.88	0.89	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
18	0.66	0.57	0.79	1.00	0.82	0.66	0.66	0.66	0.59	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
19	0.21	0.00	0.21	0.21	0.43	0.43	0.43	0.55	0.88	0.89	0.89	0.89	0.89	0.89	0.89	0.89	0.89	0.89
20	0.21	0.00	0.21	0.21	0.43	0.28	0.28	0.42	0.85	0.87	0.87	0.87	1.00	0.00	0.00	0.00	0.00	0.00
21	0.66	0.75	0.00	0.75	0.82	0.85	0.85	1.00	1.00	1.00	1.00	1.00	1.00	0.00	0.00	0.00	0.00	0.00
22	0.77	0.77	0.75	0.00	0.79	0.59	0.59	0.79	0.50	0.66	0.66	0.66	0.66	0.00	0.00	0.00	0.00	0.00
23	0.55	0.82	0.82	0.79	0.00	0.50	0.50	0.39	0.75	0.79	0.79	0.79	0.79	0.00	0.00	0.00	0.00	0.00
24	0.43	0.28	0.85	0.59	0.50	0.00	0.00	0.19	0.79	0.82	0.82	0.82	0.82	0.00	0.00	0.00	0.00	0.00
25	0.43	0.28	0.85	0.59	0.50	0.00	0.00	0.19	0.79	0.82	0.82	0.82	0.82	0.00	0.00	0.00	0.00	0.00
26	0.43	0.28	0.85	0.59	0.50	0.00	0.00	0.19	0.79	0.82	0.82	0.82	0.82	0.00	0.00	0.00	0.00	0.00
27	0.55	0.42	1.00	0.79	0.79	0.19	0.19	0.00	0.75	0.79	0.79	0.79	0.79	0.00	0.00	0.00	0.00	0.00
28	0.55	0.85	1.00	0.79	0.50	0.79	0.79	0.75	0.00	0.50	0.50	0.50	0.50	0.00	0.00	0.00	0.00	0.00
29	0.80	0.87	1.00	0.86	0.79	0.82	0.82	0.79	0.50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Anhang 16: Tanimoto-Abstandsmatrix (zu Matrix B)

MATRIX = TANIMOTO ABSTANDSMATRIX

M/M	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
31	0.89	0.87	1.00	0.66	0.79	0.82	0.82	0.82	0.79	0.50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Anhang 16: Blatt 2

Anhang 17: Abstandsmatrix mit dem Matching-Koeffizienten (Matrix B)

CLUSTERANALYSE VERFAHREN DATUM 20.04.80 SEITE 1

MATRIX = ABSTANDSMATRIX MATCHING KOEFF

M/N	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
31	9	7	5	2	4	5	5	5	4	1	0	0	0	0	0	0	0	0

Anhang 17: Blatt 2

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	
1	1.00	1.00	0.37	0.37	0.50	0.82	0.82	0.42	0.35	0.45	0.09	0.29	0.26	0.19	0.00	0.09	0.00	0.00	1
2	1.00	1.00	0.37	0.37	0.50	0.82	0.82	0.42	0.35	0.45	0.09	0.29	0.26	0.19	0.00	0.09	0.00	0.00	2
3	0.37	0.37	1.00	1.00	0.62	0.50	0.50	0.57	0.45	0.53	0.32	0.39	0.50	0.43	0.32	0.32	0.21	0.00	3
4	0.37	0.37	1.00	1.00	0.62	0.50	0.50	0.57	0.45	0.53	0.32	0.39	0.50	0.43	0.32	0.32	0.21	0.00	4
5	0.82	0.50	0.62	0.62	1.00	0.62	0.62	0.37	0.41	0.53	0.17	0.25	0.32	0.26	0.25	0.17	0.08	0.00	5
6	0.82	0.50	0.50	0.50	0.62	1.00	1.00	0.37	0.45	0.53	0.19	0.26	0.35	0.29	0.00	0.08	0.00	0.00	6
7	0.82	0.82	0.57	0.57	0.37	0.37	1.00	0.35	0.45	0.53	0.21	0.43	0.39	0.19	0.00	0.37	0.25	0.00	7
8	0.42	0.42	0.57	0.57	0.37	0.37	0.35	1.00	0.90	1.00	0.53	0.72	0.81	0.63	0.39	0.25	0.07	0.00	8
9	0.35	0.35	0.45	0.45	0.41	0.45	0.45	0.35	1.00	0.90	0.53	0.79	0.89	0.69	0.07	0.53	0.50	0.07	9
10	0.45	0.45	0.53	0.53	0.53	0.53	0.53	0.45	0.90	1.00	0.53	0.79	0.89	0.69	0.07	0.53	0.45	0.07	10
11	0.09	0.09	0.32	0.32	0.17	0.19	0.21	0.59	0.53	0.53	1.00	0.55	0.66	0.62	0.12	0.50	0.57	0.09	11
12	0.29	0.29	0.39	0.39	0.25	0.26	0.43	0.79	0.72	0.79	0.55	1.00	0.88	0.86	0.09	0.75	0.62	0.09	12
13	0.26	0.26	0.50	0.50	0.32	0.35	0.39	0.89	0.81	0.89	0.66	0.88	1.00	0.77	0.08	0.66	0.55	0.08	13
14	0.19	0.19	0.43	0.43	0.26	0.29	0.29	0.89	0.83	0.89	0.62	0.86	0.77	1.00	0.10	0.62	0.50	0.10	14
15	0.00	0.00	0.12	0.12	0.25	0.00	0.00	0.07	0.07	0.12	0.09	0.08	0.08	0.10	1.00	0.12	0.19	0.08	15
16	0.09	0.09	0.32	0.32	0.17	0.08	0.08	0.37	0.59	0.53	0.50	0.75	0.66	0.62	0.12	1.00	0.82	0.12	16
17	0.00	0.00	0.21	0.21	0.08	0.00	0.00	0.25	0.50	0.45	0.57	0.62	0.55	0.50	0.14	0.82	1.00	0.14	17
18	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.07	0.07	0.00	0.09	0.08	0.08	0.10	0.19	0.12	0.14	1.00	18
19	0.00	0.00	0.14	0.14	0.16	0.00	0.00	0.16	0.35	0.32	0.35	0.41	0.38	0.32	0.32	0.50	0.55	0.32	19
20	0.00	0.00	0.17	0.17	0.16	0.00	0.00	0.08	0.21	0.19	0.17	0.25	0.26	0.22	0.42	0.29	0.32	0.42	20
21	0.00	0.00	0.12	0.12	0.10	0.00	0.00	0.14	0.29	0.26	0.28	0.37	0.32	0.25	0.19	0.50	0.59	0.19	21
22	0.00	0.00	0.14	0.14	0.12	0.00	0.00	0.00	0.08	0.07	0.14	0.10	0.09	0.12	0.25	0.14	0.19	0.00	22
23	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.12	0.07	0.07	0.10	0.08	0.07	0.00	0.16	0.10	0.12	0.16	23
24	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.07	0.06	0.06	0.09	0.07	0.07	0.08	0.59	0.09	0.10	0.32	24
25	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.07	0.06	0.06	0.09	0.07	0.07	0.08	0.59	0.09	0.10	0.32	25
26	0.00	0.00	0.09	0.09	0.00	0.00	0.00	0.07	0.06	0.06	0.09	0.07	0.07	0.08	0.59	0.09	0.10	0.32	26
27	0.00	0.00	0.09	0.09	0.00	0.00	0.00	0.07	0.06	0.06	0.09	0.07	0.07	0.08	0.59	0.09	0.10	0.32	27
28	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.39	0.00	0.00	0.39	28
29	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	29
30	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	30
31	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	31
1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1
2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	2
3	0.14	0.14	0.17	0.17	0.12	0.14	0.14	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.00	0.09	0.00	0.00	3
4	0.14	0.14	0.17	0.17	0.12	0.14	0.14	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.00	0.09	0.00	0.00	4
5	0.14	0.14	0.16	0.16	0.12	0.14	0.14	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.00	0.09	0.00	0.00	5
6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	6
7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	7
8	0.08	0.08	0.14	0.14	0.08	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	8
9	0.35	0.21	0.29	0.29	0.08	0.07	0.07	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	9
10	0.32	0.19	0.26	0.26	0.07	0.06	0.06	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	10
11	0.35	0.17	0.28	0.28	0.10	0.09	0.09	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	11
12	0.41	0.25	0.37	0.37	0.10	0.08	0.07	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	12
13	0.38	0.22	0.32	0.32	0.09	0.07	0.07	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	13
14	0.32	0.26	0.25	0.25	0.12	0.00	0.08	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	14
15	0.32	0.32	0.42	0.42	0.16	0.59	0.59	0.39	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	15
16	0.30	0.29	0.50	0.50	0.16	0.10	0.09	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	16
17	0.35	0.32	0.50	0.50	0.16	0.10	0.10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	17
18	0.32	0.42	0.19	0.19	0.00	0.16	0.32	0.32	0.39	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	18
19	1.00	0.77	0.32	0.32	0.21	0.43	0.55	0.55	0.43	0.10	0.09	0.09	0.00	0.00	0.00	0.00	0.00	0.00	19
20	0.77	1.00	0.25	0.25	0.28	0.37	0.71	0.71	0.57	0.14	0.12	0.12	0.12	0.00	0.00	0.00	0.00	0.00	20
21	0.32	0.25	1.00	1.00	0.25	0.16	0.14	0.14	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	21
22	0.21	0.28	0.25	0.25	1.00	0.16	0.39	0.39	0.19	0.50	0.32	0.32	0.32	0.00	0.00	0.00	0.00	0.00	22
23	0.63	0.37	0.16	0.16	0.39	1.00	0.50	0.50	0.39	0.25	0.19	0.19	0.19	0.00	0.00	0.00	0.00	0.00	23
24	0.55	0.71	0.14	0.14	0.39	0.50	1.00	1.00	0.79	0.19	0.16	0.16	0.16	0.00	0.00	0.00	0.00	0.00	24
25	0.55	0.71	0.14	0.14	0.39	0.50	1.00	1.00	0.79	0.19	0.16	0.16	0.16	0.00	0.00	0.00	0.00	0.00	25
26	0.56	0.71	0.14	0.14	0.39	0.50	1.00	1.00	0.79	0.19	0.16	0.16	0.16	0.00	0.00	0.00	0.00	0.00	26
27	0.43	0.57	0.00	0.00	0.39	0.79	0.79	0.79	1.00	0.25	0.19	0.19	0.19	0.00	0.00	0.00	0.00	0.00	27
28	0.10	0.14	0.00	0.00	0.25	0.19	0.19	0.19	0.25	1.00	0.50	0.50	0.50	0.00	0.00	0.00	0.00	0.00	28
29	0.09	0.12	0.00	0.00	0.32	0.19	0.16	0.16	0.19	0.50	1.00	1.00	1.00	0.00	0.00	0.00	0.00	0.00	29
30	0.09	0.12	0.00	0.00	0.32	0.19	0.16	0.16	0.19	0.50	1.00	1.00	1.00	0.00	0.00	0.00	0.00	0.00	30
31	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	31
32	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	32
33	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	33
34	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	34
35	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	35
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	

Anhang 18: 1. Ähnlichkeitsmatrix nach Zimmermann (Matrix B)

CLUSTERANALYSE VERFAHREN

MATRIX = 1.AEHNLICHK.MASS N. ZIMMERMANN

M/N	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
31	0.09	0.12	0.00	0.32	0.19	0.16	0.16	0.16	0.19	0.50	1.00	1.00	1.00	0.00	0.00	0.00	0.00	0.00

Anhang 18: Blatt 2

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	1.00	1.00	0.50	0.59	0.79	1.00	1.00	0.59	0.79	1.00	0.19	0.59	0.59	0.39	0.00	0.19	0.00	0.00
2	1.00	1.00	0.59	0.59	0.79	1.00	1.00	0.59	0.79	1.00	0.19	0.59	0.59	0.39	0.00	0.19	0.00	0.00
3	0.59	0.59	1.00	1.00	0.82	0.66	0.66	0.79	0.82	1.00	0.50	0.66	0.82	0.66	0.32	0.50	0.39	0.00
4	0.79	0.79	1.00	1.00	0.82	0.66	0.66	0.79	0.82	1.00	0.50	0.66	0.82	0.66	0.32	0.50	0.39	0.00
5	0.79	0.79	0.82	0.82	1.00	0.82	0.82	0.59	0.82	1.00	0.32	0.50	0.57	0.42	0.00	0.16	0.00	0.00
6	1.00	1.00	0.66	0.82	1.00	1.00	1.00	0.59	0.82	1.00	0.32	0.50	0.66	0.50	0.00	0.16	0.00	0.00
7	1.00	1.00	0.66	0.82	1.00	1.00	1.00	0.59	0.82	1.00	0.32	0.50	0.66	0.50	0.00	0.16	0.00	0.00
8	0.59	0.59	0.79	0.59	0.59	0.59	0.59	1.00	0.79	1.00	0.39	0.79	0.79	0.39	0.00	0.59	0.39	0.00
9	0.79	0.79	0.82	0.82	0.82	0.82	0.82	0.79	1.00	1.00	0.79	1.00	1.00	0.79	0.32	1.00	0.32	0.32
10	1.00	1.00	1.00	1.00	0.82	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.32	1.00	0.32	0.32
11	0.19	0.19	0.50	0.50	0.32	0.32	0.32	0.39	1.00	1.00	1.00	0.82	1.00	0.85	0.32	1.00	0.32	0.32
12	0.59	0.59	0.66	0.66	0.42	0.50	0.50	0.79	1.00	1.00	0.82	1.00	1.00	1.00	0.32	1.00	0.32	0.32
13	0.59	0.59	0.82	0.82	0.57	0.66	0.66	0.79	1.00	1.00	1.00	1.00	1.00	1.00	0.32	1.00	0.32	0.32
14	0.39	0.39	0.66	0.66	0.42	0.50	0.50	0.39	1.00	1.00	0.82	1.00	1.00	1.00	0.32	0.82	0.32	0.32
15	0.00	0.00	0.32	0.32	0.00	0.00	0.00	0.32	0.32	0.32	0.32	0.32	0.32	0.32	1.00	0.32	1.00	1.00
16	0.19	0.19	0.50	0.50	0.32	0.32	0.32	0.39	1.00	1.00	0.82	1.00	1.00	1.00	0.32	1.00	0.32	0.32
17	0.00	0.00	0.39	0.39	0.19	0.00	0.00	0.32	0.32	0.32	0.32	0.32	0.32	0.32	1.00	0.32	1.00	1.00
18	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.32	0.32	0.32	0.32	0.32	0.32	0.32	1.00	0.32	1.00	1.00
19	0.00	0.00	0.32	0.32	0.28	0.00	0.00	0.39	0.55	0.55	0.66	0.62	0.55	0.57	1.00	0.32	0.82	0.32
20	0.00	0.00	0.32	0.32	0.28	0.00	0.00	0.39	0.42	0.42	0.32	0.42	0.42	0.42	1.00	0.32	0.82	0.32
21	0.00	0.00	0.32	0.32	0.32	0.00	0.00	0.32	1.00	1.00	0.66	1.00	1.00	1.00	0.32	1.00	0.32	0.32
22	0.00	0.00	0.50	0.50	0.32	0.00	0.00	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.32	0.50	0.50	0.50
23	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.00	0.25	0.25	0.25
24	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
25	0.00	0.00	0.19	0.19	0.39	0.00	0.00	0.19	0.19	0.19	0.19	0.19	0.19	0.19	0.32	0.19	0.19	0.19
26	0.00	0.00	0.19	0.19	0.39	0.00	0.00	0.19	0.19	0.19	0.19	0.19	0.19	0.19	0.32	0.19	0.19	0.19
27	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
28	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
29	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
30	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
31	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
3	0.32	0.32	0.32	0.32	0.19	0.19	0.19	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	0.32	0.32	0.32	0.32	0.19	0.19	0.19	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	0.28	0.28	0.32	0.32	0.39	0.39	0.39	0.25	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
8	0.39	0.39	0.19	0.32	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
9	0.55	0.55	0.42	1.00	0.50	0.25	0.19	0.19	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
10	0.55	0.55	0.42	1.00	0.50	0.25	0.19	0.19	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
11	0.68	0.68	0.32	0.66	0.50	0.25	0.19	0.19	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
12	0.82	0.82	0.42	1.00	0.50	0.25	0.19	0.19	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
13	0.55	0.55	0.42	1.00	0.50	0.25	0.19	0.19	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
14	0.57	0.57	0.42	0.66	0.50	0.00	0.19	0.19	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
15	1.00	1.00	1.00	1.00	0.50	0.50	0.19	0.19	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
16	0.82	0.82	0.50	1.00	0.50	0.25	0.19	0.19	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
17	1.00	1.00	0.59	1.00	0.50	0.25	0.19	0.19	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
18	1.00	1.00	1.00	1.00	0.50	0.25	0.19	0.19	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
19	1.00	1.00	1.00	1.00	0.50	0.25	0.19	0.19	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
20	1.00	1.00	1.00	1.00	0.50	0.25	0.19	0.19	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
21	1.00	1.00	0.68	1.00	0.50	0.32	0.32	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
22	1.00	1.00	0.50	1.00	0.50	0.50	0.50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
23	1.00	1.00	0.75	0.75	0.75	0.75	0.75	0.75	1.00	1.00	0.50	0.50	0.50	0.50	0.00	0.00	0.00	0.00
24	1.00	1.00	0.75	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.50	0.50	0.50	0.50	0.00	0.00	0.00	0.00
25	1.00	1.00	0.75	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.50	0.50	0.50	0.50	0.00	0.00	0.00	0.00
26	1.00	1.00	0.75	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.50	0.50	0.50	0.50	0.00	0.00	0.00	0.00
27	1.00	1.00	0.50	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.50	0.50	0.50	0.50	0.00	0.00	0.00	0.00
28	1.00	1.00	0.50	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.50	0.50	0.50	0.50	0.00	0.00	0.00	0.00
29	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.00	0.00	0.00	0.00
30	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.00	0.00	0.00	0.00

Anhang 19: 2. Ähnlichkeitsmaß nach Zimmermann
(Matrix B)

CLUSTERANALYSE VERFAHREN

MATRIX = 2.AEHNLICHK.MASS N. ZIMMERMANN

M/N	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
31	0.50	0.50	0.00	0.50	0.50	0.50	0.50	0.50	0.50	1.00	1.00	1.00	1.00	0.00	0.00	0.00	0.00	0.00

Anhang 19: Blatt 2

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	1.00	1.00	0.50	0.50	0.57	0.82	0.82	0.59	0.39	0.45	0.16	0.37	0.32	0.28	0.00	0.16	0.00	0.00
2	1.00	1.00	0.50	0.50	0.57	0.82	0.82	0.59	0.39	0.45	0.16	0.37	0.32	0.28	0.00	0.16	0.00	0.00
3	0.50	0.50	1.00	1.00	0.71	0.66	0.66	0.50	0.53	0.53	0.50	0.50	0.55	0.57	0.16	0.50	0.32	0.00
4	0.50	0.50	1.00	1.00	0.71	0.66	0.66	0.50	0.53	0.53	0.50	0.50	0.55	0.57	0.16	0.50	0.32	0.00
5	0.57	0.57	0.71	0.71	1.00	0.71	0.71	0.50	0.53	0.53	0.28	0.37	0.43	0.42	0.28	0.00	0.14	0.00
6	0.82	0.82	0.66	0.66	0.71	1.00	1.00	0.50	0.53	0.53	0.32	0.37	0.43	0.42	0.00	0.16	0.00	0.00
7	0.82	0.82	0.66	0.66	0.71	1.00	1.00	0.50	0.53	0.53	0.32	0.37	0.43	0.42	0.00	0.16	0.00	0.00
8	0.59	0.59	0.50	0.50	0.50	0.50	0.50	1.00	0.39	0.45	0.16	0.37	0.32	0.28	0.00	0.16	0.00	0.00
9	0.39	0.39	0.53	0.53	0.53	0.53	0.53	0.39	1.00	0.90	0.59	0.79	0.89	0.69	0.09	0.59	0.50	0.09
10	0.45	0.45	0.53	0.53	0.53	0.53	0.53	0.45	0.90	1.00	0.53	0.72	0.81	0.63	0.08	0.53	0.45	0.08
11	0.16	0.16	0.50	0.50	0.28	0.32	0.32	0.32	0.59	0.53	1.00	0.62	0.66	0.71	0.16	0.66	0.66	0.00
12	0.37	0.37	0.50	0.50	0.37	0.37	0.37	0.50	0.79	0.72	0.62	1.00	0.88	0.75	0.12	0.75	0.62	0.12
13	0.32	0.32	0.55	0.55	0.43	0.43	0.43	0.43	0.89	0.81	0.66	0.88	1.00	0.77	0.10	0.66	0.55	0.10
14	0.28	0.28	0.57	0.57	0.42	0.42	0.42	0.28	0.69	0.63	0.71	0.75	0.77	1.00	0.14	0.71	0.57	0.14
15	0.00	0.00	0.16	0.16	0.28	0.00	0.00	0.00	0.09	0.08	0.16	0.12	0.10	0.14	1.00	0.16	0.19	0.32
16	0.16	0.16	0.50	0.50	0.50	0.50	0.50	0.39	0.59	0.53	0.66	0.75	0.66	0.71	0.16	1.00	0.82	0.16
17	0.00	0.00	0.32	0.32	0.14	0.00	0.00	0.00	0.50	0.45	0.66	0.62	0.55	0.57	0.19	0.82	1.00	0.19
18	0.00	0.00	0.32	0.32	0.14	0.00	0.00	0.00	0.50	0.45	0.66	0.62	0.55	0.57	0.19	0.82	1.00	0.19
19	0.00	0.00	0.28	0.28	0.28	0.00	0.00	0.00	0.09	0.08	0.16	0.12	0.10	0.14	0.32	0.16	0.19	1.00
20	0.00	0.00	0.28	0.28	0.28	0.00	0.00	0.00	0.09	0.08	0.16	0.12	0.10	0.14	0.32	0.16	0.19	1.00
21	0.00	0.00	0.16	0.16	0.14	0.00	0.00	0.00	0.09	0.08	0.16	0.12	0.10	0.14	0.59	0.16	0.19	0.39
22	0.00	0.00	0.16	0.16	0.14	0.00	0.00	0.00	0.09	0.08	0.16	0.12	0.10	0.14	0.59	0.16	0.19	0.39
23	0.00	0.00	0.16	0.16	0.14	0.00	0.00	0.00	0.09	0.08	0.16	0.12	0.10	0.14	0.59	0.16	0.19	0.39
24	0.00	0.00	0.16	0.16	0.14	0.00	0.00	0.00	0.09	0.08	0.16	0.12	0.10	0.14	0.59	0.16	0.19	0.39
25	0.00	0.00	0.16	0.16	0.28	0.00	0.00	0.00	0.09	0.08	0.16	0.12	0.10	0.14	0.59	0.16	0.19	0.39
26	0.00	0.00	0.16	0.16	0.28	0.00	0.00	0.00	0.09	0.08	0.16	0.12	0.10	0.14	0.59	0.16	0.19	0.39
27	0.00	0.00	0.16	0.16	0.28	0.00	0.00	0.00	0.09	0.08	0.16	0.12	0.10	0.14	0.59	0.16	0.19	0.39
28	0.00	0.00	0.00	0.00	0.14	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.50	0.00	0.00	0.50
29	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.50	0.00	0.00	0.50
30	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.50	0.00	0.00	0.50
31	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.50	0.00	0.00	0.50
1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	0.21	0.21	0.28	0.16	0.14	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	0.21	0.21	0.28	0.16	0.14	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
8	0.21	0.21	0.14	0.14	0.14	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
9	0.50	0.50	0.29	0.29	0.29	0.09	0.09	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
10	0.45	0.45	0.26	0.26	0.26	0.08	0.08	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
11	0.45	0.45	0.28	0.32	0.16	0.16	0.16	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
12	0.55	0.55	0.32	0.32	0.10	0.10	0.10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
13	0.55	0.55	0.32	0.32	0.10	0.10	0.10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
14	0.43	0.43	0.28	0.32	0.14	0.00	0.14	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
15	0.32	0.32	0.42	0.32	0.25	0.00	0.16	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
16	0.55	0.55	0.42	0.50	0.16	0.16	0.16	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
17	0.42	0.42	0.42	0.32	0.19	0.19	0.19	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
18	0.32	0.32	0.42	0.32	0.20	0.39	0.39	0.50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
19	1.00	1.00	0.77	0.77	0.71	0.43	0.55	0.55	0.43	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
20	0.77	1.00	0.28	0.28	0.42	0.71	0.71	0.57	0.14	0.14	0.14	0.14	0.00	0.00	0.00	0.00	0.00	0.00
21	0.32	0.28	1.00	0.32	0.25	0.19	0.19	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
22	0.21	0.28	0.32	1.00	0.25	0.39	0.39	0.25	0.50	0.50	0.50	0.50	0.50	0.50	0.00	0.00	0.00	0.00
23	0.42	0.42	0.25	0.25	1.00	0.59	0.59	0.75	0.25	0.25	0.25	0.25	0.25	0.25	0.00	0.00	0.00	0.00
24	0.55	0.71	0.39	0.39	0.59	1.00	1.00	1.00	0.19	0.19	0.19	0.19	0.19	0.19	0.00	0.00	0.00	0.00
25	0.55	0.71	0.39	0.39	0.59	1.00	1.00	1.00	0.19	0.19	0.19	0.19	0.19	0.19	0.00	0.00	0.00	0.00
26	0.43	0.57	0.39	0.25	0.39	0.59	0.79	1.00	0.25	0.25	0.25	0.25	0.25	0.25	0.00	0.00	0.00	0.00
27	0.43	0.57	0.39	0.25	0.39	0.59	0.79	1.00	0.25	0.25	0.25	0.25	0.25	0.25	0.00	0.00	0.00	0.00
28	0.10	0.14	0.50	0.50	0.25	0.19	0.19	0.25	1.00	1.00	1.00	1.00	1.00	1.00	0.00	0.00	0.00	0.00
29	0.10	0.14	0.50	0.50	0.25	0.19	0.19	0.25	1.00	1.00	1.00	1.00	1.00	1.00	0.00	0.00	0.00	0.00
30	0.10	0.14	0.50	0.50	0.25	0.19	0.19	0.25	1.00	1.00	1.00	1.00	1.00	1.00	0.00	0.00	0.00	0.00

Anhang 2o: 3. Ähnlichkeitsmaß n. Zimmermann
(Matrix B)

CLUSTERANALYSE

MATRIX = 3.AEHNLICHK.MASS N. ZIMMERMANN

M/N	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
31	0.10	0.14	0.00	0.50	0.25	0.19	0.19	0.19	0.25	0.50	1.00	1.00	1.00	0.00	0.00	0.00	0.00	0.00

MATRIX = AEHNLICHKEITSMATRIX N.FREUD.

M/N	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	31	29	28	16	12	7	25	18	18	20	26	11	16	11	14	25	8	7
2	20	31	28	18	10	5	25	18	20	22	24	9	14	9	16	25	6	10
3	28	26	31	13	15	10	22	15	15	17	27	14	19	14	11	24	11	14
4	16	18	13	31	17	12	20	25	27	25	17	18	17	16	21	18	14	14
5	12	10	15	17	31	22	12	13	15	13	17	28	19	28	17	14	16	16
6	17	5	10	12	22	31	9	12	12	10	12	25	22	25	12	9	14	14
7	25	25	22	20	12	9	31	20	24	20	26	13	18	13	14	25	9	14
8	18	18	15	25	15	12	20	31	27	25	17	16	17	16	19	18	12	16
9	18	20	15	27	15	12	24	27	31	27	19	16	19	16	21	22	12	16
10	20	22	17	25	15	10	20	25	27	31	15	14	15	14	23	22	10	12
11	26	24	27	17	17	12	26	17	19	15	31	16	21	16	11	24	14	14
12	11	9	14	18	28	25	13	16	16	14	16	31	20	29	9	13	8	8
13	16	14	19	17	19	22	18	17	19	15	21	20	31	20	14	18	11	11
14	11	9	14	16	28	25	13	16	16	14	16	29	20	31	16	13	8	8
15	14	16	11	21	17	12	14	19	21	23	11	14	9	16	31	18	8	8
16	25	25	24	18	14	9	25	18	22	22	24	13	18	13	18	31	7	8

Anhang 21: Ähnlichkeitsmatrix nach dem Verfahren von Freudenthaler und Maier für die transponierte Matrix B

CLUSTERANALYSE VERFAHREN DATUM 28.04.80 SEITE 9

MATRIX = BAUM

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	16	1	2	3	7	9	10	0	0	0	0	0	0	0	0	0	0	0
2	16	2	3	7	9	10	0	0	0	0	0	0	0	0	0	0	0	0
3	16	3	7	9	10	0	0	0	0	0	0	0	0	0	0	0	0	0
4	16	7	9	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	16	9	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	16	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	16	11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	4	8	4	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	5	6	12	14	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	13	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	15	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Anhang 22: Baumbildung auf dem Klassifikationsniveau 22
für die transponierte Matrix B

Anhang 23: Baumbildung auf dem Klassifikationsniveau 16
für die transponierte Matrix B

MATRIX = BAUM

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	5	12	14															
2	1	2	0															
3	1	3	0															
4	1	0	0															
5	6	0	0															
6	7	0	0															
7	8	0	0															
8	9	0	0															
9	10	0	0															
10	11	0	0															
11	13	0	0															
12	15	0	0															
13	16	0	0															

Anhang 24: Baumbildung auf dem Klassifikationsniveau 28
für die transponierte Matrix B

Anhang 25: Matrix D
vgl. Döringer, a.a.O., S.94

CLUSTERANALYSE VERFAHREN — DATUM 28.04.80 SEITE

MATRIX = AEHNLICHKEITSMATRIX N.FREUD.

M/N	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	15	6	7	10	8	8	8	6	6	6	6	6	4	2	3	3	2	4
2	6	15	14	7	9	9	11	13	13	13	13	13	5	9	10	10	11	13
3	7	14	15	6	8	8	10	12	12	12	12	12	4	8	9	9	10	12
4	10	7	6	15	13	13	7	7	7	7	7	7	9	7	8	8	7	9
5	8	9	8	13	15	15	9	9	9	9	9	9	7	7	8	8	9	11
6	8	9	8	13	15	15	9	9	9	9	9	9	7	7	8	8	9	11
7	8	11	10	7	9	9	15	13	13	13	13	13	3	7	8	8	9	11
8	6	13	12	7	9	9	13	15	15	15	15	15	5	9	10	10	11	13
9	6	13	12	7	9	9	13	15	15	15	15	15	5	9	10	10	11	13
10	6	13	12	7	9	9	13	15	15	15	15	15	5	9	10	10	11	13
11	6	13	12	7	9	9	13	15	15	15	15	15	5	9	10	10	11	13
12	6	13	12	7	9	9	13	15	15	15	15	15	5	9	10	10	11	13
13	4	5	4	9	7	7	3	5	5	5	5	5	15	11	10	10	9	7
14	2	9	8	7	7	7	7	9	9	9	9	9	11	15	12	12	11	11
15	3	10	9	8	8	8	8	10	10	10	10	10	10	12	15	15	14	12
16	3	10	9	8	8	8	8	10	10	10	10	10	10	12	15	15	14	12
17	2	11	10	7	9	9	9	11	11	11	11	11	9	11	14	14	15	13
18	4	13	12	9	11	11	11	13	13	13	13	13	7	11	12	12	13	15
19	2	11	10	7	9	9	9	11	11	11	11	11	9	11	14	14	15	13
20	2	11	10	7	9	9	9	11	11	11	11	11	9	13	10	10	11	13
21	2	11	10	7	9	9	9	11	11	11	11	11	9	13	10	10	11	13
22	2	11	10	7	9	9	9	11	11	11	11	11	9	13	10	10	11	13
1	2	2	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	11	11	11	11	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	10	10	10	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	7	7	7	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	9	9	9	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	9	9	9	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	9	9	9	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	11	11	11	11	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	11	11	11	11	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	11	11	11	11	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	11	11	11	11	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	11	11	11	11	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	9	9	9	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	11	13	13	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	14	10	10	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	14	10	10	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	15	11	11	11	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	13	13	13	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	15	11	11	11	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	11	15	15	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21	11	15	15	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	11	15	15	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Spalte 11 - 22

Anhang 26: Ähnlichkeitsmatrix nach dem Verfahren von Freudenthaler und Maier für die Matrix D

MATRIX D AEHNLICHKEITSMATRIX N.FREUD.

M/N	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	22	19	20	19	20	18	18	14	19	15	16	12	19	16	16			
2	19	22	17	20	17	21	17	15	16	14	15	15	18	15	15			
3	20	17	22	17	22	16	20	12	17	13	14	10	21	14	14			
4	19	20	17	22	17	19	19	17	16	14	15	15	18	15	15			
5	20	17	22	17	22	16	20	12	17	13	14	10	21	14	14			
6	18	21	16	19	16	22	16	15	15	13	15	15	18	15	15			
7	18	17	20	19	20	16	22	14	15	11	14	10	21	14	14			
8	14	15	12	17	12	14	14	22	11	9	12	12	17	12	12			
9	19	16	17	16	17	15	15	11	22	18	15	12	16	10	10			
10	15	14	13	14	13	13	11	9	18	22	15	7	12	21	15			
11	16	15	14	15	14	14	12	10	15	15	22	8	13	14	22			
12	12	15	10	15	10	16	12	20	9	7	8	22	11	8	13			
13	19	18	21	18	21	17	21	13	16	12	13	8	22	13	13			
14	16	15	14	15	14	14	12	10	17	21	14	8	13	22	14			
15	15	16	15	16	15	16	15	17	15	15	22	8	13	14	22			
16	16	15	14	15	14	14	12	10	17	21	14	8	13	22	14			
17	15	14	13	14	13	13	11	9	18	22	15	7	12	21	15			
18	14	15	12	15	12	14	12	10	15	15	22	8	13	14	22			

Anhang 27: Ähnlichkeitsmatrix nach dem Verfahren von Freudenthaler und Maier für die transponierte Matrix D

CLUSTERANALYSE VERFAHREN DATUM 28.04.80 SEITE 8

MATRIX = BAUM

M/N	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	18	2	3	7	8	9	10	11	12	15	16	17	19	20	21	22		
2	18	3	5	6	0	0	0	0	0	0	0	0	0	0	0	0		
3	18	5	6	0	0	0	0	0	0	0	0	0	0	0	0	0		
4	18	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
5	18	7	8	9	10	11	12	15	16	17	19	20	21	22	0	0		
6	18	8	9	10	11	12	15	16	17	19	20	21	22	0	0	0		
7	18	9	10	11	12	15	16	17	19	20	21	22	0	0	0	0		
8	18	10	11	12	15	16	17	19	20	21	22	0	0	0	0	0		
9	18	11	12	15	16	17	19	20	21	22	0	0	0	0	0	0		
10	18	12	15	16	17	19	20	21	22	0	0	0	0	0	0	0		
11	18	14	15	16	17	19	20	21	22	0	0	0	0	0	0	0		
12	14	15	16	0	0	0	0	0	0	0	0	0	0	0	0	0		
13	13	14	15	16	0	0	0	0	0	0	0	0	0	0	0	0		

Anhang 28: Baumbildung auf dem Klassifikationsniveau 10
(Matrix D)

Anhang 31: Baumbildung auf dem Klassifikationsniveau 2o
für die transponierte Matrix D

Anhang 32: Baumbildung auf dem Klassifikationsniveau 22
für die transponierte Matrix D

MATRIX = INPUT DATENMATRIX

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	0	1	1	1	0	1	0	1	1	0	1	1	0					
2	0	1	1	1	0	1	0	1	1	0	1	1	0					
3	0	1	1	1	0	1	0	1	1	0	1	1	0					
4	0	1	1	1	0	1	0	1	1	0	1	1	0					
5	0	1	1	1	0	1	0	1	1	0	1	1	0					
6	0	1	1	1	0	1	0	1	1	0	1	1	0					
7	0	1	1	1	0	1	0	1	1	0	1	1	0					
8	0	1	1	1	0	1	0	1	1	0	1	1	0					
9	0	1	1	1	0	1	0	1	1	0	1	1	0					
10	0	1	1	1	0	1	0	1	1	0	1	1	0					
11	0	1	1	1	0	1	0	1	1	0	1	1	0					
12	0	1	1	1	0	1	0	1	1	0	1	1	0					
13	0	1	1	1	0	1	0	1	1	0	1	1	0					
14	0	1	1	1	0	1	0	1	1	0	1	1	0					
15	0	1	1	1	0	1	0	1	1	0	1	1	0					
16	0	1	1	1	0	1	0	1	1	0	1	1	0					
17	0	1	1	1	0	1	0	1	1	0	1	1	0					
18	0	1	1	1	0	1	0	1	1	0	1	1	0					
19	0	1	1	1	0	1	0	1	1	0	1	1	0					
20	0	1	1	1	0	1	0	1	1	0	1	1	0					
21	0	1	1	1	0	1	0	1	1	0	1	1	0					
22	0	1	1	1	0	1	0	1	1	0	1	1	0					
23	0	1	1	1	0	1	0	1	1	0	1	1	0					
24	0	1	1	1	0	1	0	1	1	0	1	1	0					

Anhang 33: Matrix K, vgl. Koreimann, D.S.: Anwendungsanalyse, a.a.O., S.3

MATRIX = AEHNLICHKEITSMATRIX N.FREUD.

Anhang 34: Ähnlichkeitsmatrix nach dem Verfahren von Freudenthaler und Maier für die Matrix K

MATRIX = BAUM

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	23	2	8	9	10	12	17	18	19	20	22							
2	23	4	8	9	10	12	17	18	19	20	0							
3	23	6	9	10	12	17	18	19	20	22								
4	23	8	9	10	12	17	18	19	20	0								
5	23	9	10	12	17	18	19	20	22									
6	23	10	12	17	18	19	20	22	0									
7	23	12	17	18	19	20	22	0										
8	23	13	19	20	22	0												
9	23	14	16	17	18	19	20	22	0									
10	23	16	17	18	19	20	22	0										
11	23	17	18	19	20	22	0											
12	23	18	19	20	22	0												
13	23	19	20	22	0													
14	23	20	22	0														
15	23	21	22	0														
16	1	2	3	0														
17	5	4	14	15	16	0												
18	7	11	24	0														
19	7	12	18	0														
20	7	18	0															
21	7	24	0															

Anhang 35: Baumbildung auf dem Klassifikationsniveau 9
(Matrix K)

MATRIX = AEHNLICHKEITSMATRIX N.FREUD.

M/N	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	24	12	13	17	11	11	12	9	10	12	12	12	15				
2	12	24	23	19	11	10	14	13	18	18	20	20	17				
3	13	23	24	18	10	20	13	14	17	15	19	19	16				
4	17	19	18	24	12	16	13	12	15	15	19	17	18				
5	11	11	10	12	24	10	15	16	15	19	15	15	18				
6	11	10	20	16	10	24	11	12	13	13	15	17	12				
7	12	14	15	13	15	11	24	17	14	16	16	14	19				
8	9	13	14	12	16	12	17	24	19	17	13	15	14				
9	10	18	17	15	15	15	14	19	24	16	16	16	15				
10	12	18	15	15	19	13	16	17	18	24	18	20	21				
11	12	16	15	15	15	15	14	16	16	18	24	20	19				
12	12	20	19	17	15	17	15	15	20	20	20	24	24				
13	15	17	16	18	18	12	19	14	15	21	21	19					

Anhang 36: Ähnlichkeitsmatrix nach dem Verfahren von Freudenthaler und Maier für die transponierte Matrix K

Anhang 37: Baumbildung auf dem Klassifikationsniveau 17
für die transponierte Matrix K

Adjazenzmatrix für das erste Beispiel:

i \ j	1	2	3	4	5
1	0	1	1	1	1
2	0	0	1	1	0
3	0	1	0	1	0
4	0	0	0	0	0
5	0	0	0	0	0

gewählte Gewichtungen

erste Gewichtung:

i \ j	1	2	3	4	5
1	0	1	2	3	0
2	0	0	0	4	0
3	0	0	0	5	0
4	0	0	0	0	0
5	0	0	0	0	0

zweite Gewichtung:

i \ j	1	2	3	4	5
1	0	1	0	3	0
2	0	0	0	2	0
3	0	0	0	0	0
4	0	0	0	0	0
5	0	0	0	0	0

Anhang 38: Erstes Beispiel, Berechnung nach dem Verfahren von Vetter

Beide Gewichtungen liefern ein Ergebnis, das der Lösung x_{51} auf S.74 entspricht:

Zwischenergebnis inverse Abbildungen sind noch nicht zusammengefaßt:

i \ j	1	2	3	4	5
1	0	0	1	0	1
2	0	0	1	0	0
3	0	1	0	1	0
4	0	0	0	0	0
5	0	0	0	0	0

Ergebnis, wenn inverse Abbildungen zusammengefaßt sind:

i \ j	1	2	3	4	5
1	0	0	1	0	1
2	0	1	1	1	0
3	0	0	0	0	0
4	0	0	0	0	0
5	0	0	0	0	0

Anhang 39: Erstes Beispiel, Berechnung nach dem Verfahren von Vetter

Dritte Gewichtung für das erste Beispiel:

i \ j	1	2	3	4	5
1	0	1	0	2	0
2	0	0	0	0	0
3	0	0	0	3	0
4	0	0	0	0	0
5	0	0	0	0	0

Das Zwischenergebnis, bei dem inverse Abbildungen noch nicht zusammengefaßt sind, entspricht der Lösung x_{52} auf S.74:

i \ j	1	2	3	4	5
1	0	0	1	0	1
2	0	0	1	1	0
3	0	1	0	0	0
4	0	0	0	0	0
5	0	0	0	0	0

Ergebnis, wenn inverse Abbildungen zusammengefaßt sind:

i \ j	1	2	3	4	5
1	0	0	1	0	1
2	0	1	1	1	0
3	0	0	0	0	0
4	0	0	0	0	0
5	0	0	0	0	0

Anhang 40: Erstes Beispiel,
Berechnung nach dem Verfahren von Vetter

Vierte Gewichtung für das erste Beispiel:

i \ j	1	2	3	4	5
1	0	0	1	2	0
2	0	0	0	3	0
3	0	0	0	0	0
4	0	0	0	0	0
5	0	0	0	0	0

Das Zwischenergebnis, bei dem inverse Abbildungen noch nicht zusammengefaßt sind entspricht der Lösung x_{53} auf S.74:

i \ j	1	2	3	4	5
1	0	1	0	0	1
2	0	0	1	0	0
3	0	1	0	1	0
4	0	0	0	0	0
5	0	0	0	0	0

Ergebnis, wenn inverse Abbildungen zusammengefaßt sind:

i \ j	1	2	3	4	5
1	0	1	0	0	1
2	0	1	1	1	0
3	0	0	0	0	0
4	0	0	0	0	0
5	0	0	0	0	0

Anhang 41: Erstes Beispiel,
Berechnung nach dem Verfahren von Vetter

Gewichtungen, die alle zu einem Ergebnis führen, das der Lösung x_{54} auf S. 74 entspricht:

fünfte Gewichtung:

i \ j	1	2	3	4	5
1	0	5	4	1	0
2	0	0	0	3	0
3	0	0	0	2	0
4	0	0	0	0	0
5	0	0	0	0	0

sechste Gewichtung:

i \ j	1	2	3	4	5
1	0	5	3	2	0
2	0	0	0	4	0
3	0	0	0	1	0
4	0	0	0	0	0
5	0	0	0	0	0

siebte Gewichtung:

i \ j	1	2	3	4	5
1	0	5	2	3	0
2	0	0	0	4	0
3	0	0	0	1	0
4	0	0	0	0	0
5	0	0	0	0	0

achte Gewichtung:

i \ j	1	2	3	4	5
1	0	5	4	3	0
2	0	0	0	2	0
3	0	0	0	1	0
4	0	0	0	0	0
5	0	0	0	0	0

Anhang 42: Erstes Beispiel,
Berechnung nach dem Verfahren von Vetter

Gewichtungen, die ebenfalls alle zu dem Ergebnis führen, das der Lösung x_{54} auf S. 74 entspricht:

neunte Gewichtung:

i \ j	1	2	3	4	5
1	0	0	1	2	0
2	0	0	0	0	0
3	0	0	0	3	0
4	0	0	0	0	0
5	0	0	0	0	0

zehnte Gewichtung:

i \ j	1	2	3	4	5
1	0	0	2	1	0
2	0	0	0	0	0
3	0	0	0	3	0
4	0	0	0	0	0
5	0	0	0	0	0

elfte Gewichtung:

i \ j	1	2	3	4	5
1	0	0	1	3	0
2	0	0	0	0	0
3	0	0	0	2	0
4	0	0	0	0	0
5	0	0	0	0	0

zwölfte Gewichtung:

i \ j	1	2	3	4	5
1	0	5	4	2	0
2	0	0	0	3	0
3	0	0	0	1	0
4	0	0	0	0	0
5	0	0	0	0	0

Anhang 43: Erstes Beispiel,
Berechnung nach dem Verfahren von Vetter

Ergebnis mit den Gewichtungen aus den Anhängen 42 und 43.
Das Ergebnis ist in allen diesen Fällen dasselbe.

Zwischenergebnis, inverse Abbildungen sind noch nicht zusammengefaßt:

i \ j	1	2	3	4	5
1	0	1	0	0	1
2	0	0	1	1	0
3	0	1	0	0	0
4	0	0	0	0	0
5	0	0	0	0	0

Ergebnis, bei dem inverse Abbildungen zusammengefaßt wurden:

i \ j	1	2	3	4	5
1	0	1	0	0	1
2	0	1	1	1	0
3	0	0	0	0	0
4	0	0	0	0	0
5	0	0	0	0	0

Anhang 44: Erstes Beispiel,
Berechnung nach dem Verfahren von Vetter

Dreizehnte Gewichtung:

i \ j	1	2	3	4	5
1	0	1	0	0	0
2	0	0	2	0	0
3	0	0	0	3	0
4	0	0	0	0	0
5	0	0	0	0	0

Diese Gewichtung liefert im Ergebnis eine überflüssige Relation (Zeile 1 und Spalte 4), da der Knoten 4 auch über den Knoten 3 und 2 erreicht werden kann. Dabei geht die Erreichbarkeit von Knoten 1 nach 2 verloren.

Ergebnis, bei dem inverse Abbildungen noch nicht zusammengefaßt wurden:

i \ j	1	2	3	4	5
1	0	0	1	1	1
2	0	0	1	1	0
3	0	1	0	0	0
4	0	0	0	0	0
5	0	0	0	0	0

Anhang 45: Erstes Beispiel,
Berechnung nach dem Verfahren von Vetter

Adjazenzmatrix für das zweite Beispiel:

i \ j	1	2	3	4	5	6	7	8	9
1	0	0	0	0	0	0	0	0	0
2	1	0	0	0	0	1	0	0	0
3	1	1	0	0	1	1	0	0	0
4	0	0	1	0	0	0	1	1	0
5	0	0	0	0	0	0	0	0	0
6	1	1	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	1
8	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	1	0	0

erste Gewichtung:-

i \ j	1	2	3	4	5	6	7	8	9
1	0	0	0	0	0	0	0	0	0
2	3	0	0	0	0	0	0	0	0
3	1	4	0	0	0	2	0	0	0
4	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0
6	5	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0

Zwischenergebnis, bei dem inverse Abbildungen noch nicht zusammengefaßt wurden:

i \ j	1	2	3	4	5	6	7	8	9
1	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	1	0	0	0
3	0	1	0	0	1	0	0	0	0
4	0	0	1	0	0	0	1	1	0
5	0	0	0	0	0	0	0	0	0
6	1	1	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	1
8	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	1	0	0

Ergebnis, wenn inverse Abbildungen zusammengefaßt sind:

i \ j	1	2	3	4	5	6	7	8	9
1	0	0	0	0	0	0	0	0	0
2	1	1	0	0	0	1	0	0	0
3	0	1	0	0	1	0	0	0	0
4	0	0	1	0	0	0	1	1	0
5	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	1	0	1
8	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0

Anhang 46: Zweites Beispiel,
Berechnung nach dem Verfahren von Vetter

Zweite Gewichtung:

i \ j	1	2	3	4	5	6	7	8	9
1	0	0	0	0	0	0	0	0	0
2	5	0	0	0	0	0	0	0	0
3	1	4	0	0	0	2	0	0	0
4	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0
6	3	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0

Zwischenergebnis, bei dem inverse Abbildungen noch nicht zusammengefaßt wurden:

i \ j	1	2	3	4	5	6	7	8	9
1	0	0	0	0	0	0	0	0	0
2	1	0	0	0	0	1	0	0	0
3	0	1	0	0	1	0	0	0	0
4	0	0	1	0	0	0	1	1	0
5	0	0	0	0	0	0	0	0	0
6	0	1	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	1
8	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	1	0	0

Ergebnis, wenn inverse Abbildungen zusammengefaßt sind:

i \ j	1	2	3	4	5	6	7	8	9
1	0	0	0	0	0	0	0	0	0
2	1	1	0	0	0	1	0	0	0
3	0	1	0	0	1	0	0	0	0
4	0	0	1	0	0	0	1	1	0
5	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	1	0	1
8	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0

Anhang 47: Zweites Beispiel,
Berechnung nach dem Verfahren von Vetter

Dritte Gewichtung:

i \ j	1	2	3	4	5	6	7	8	9
1	0	0	0	0	0	0	0	0	0
2	3	0	0	0	0	0	0	0	0
3	5	1	0	0	0	4	0	0	0
4	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0
6	2	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0

Zwischenergebnis, bei dem inverse Abbildungen noch nicht zusammengefaßt wurden:

i \ j	1	2	3	4	5	6	7	8	9
1	0	0	0	0	0	0	0	0	0
2	1	0	0	0	0	1	0	0	0
3	1	0	0	0	1	1	0	0	0
4	0	0	1	0	0	0	1	1	0
5	0	0	0	0	0	0	0	0	0
6	0	1	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	1
8	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	1	0	0

Ergebnis, wenn inverse Abbildungen zusammengefaßt sind:

i \ j	1	2	3	4	5	6	7	8	9
1	0	0	0	0	0	0	0	0	0
2	1	1	0	0	0	1	0	0	0
3	1	0	0	0	1	1	0	0	0
4	0	0	1	0	0	0	1	1	0
5	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	1	0	1
8	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0

Anhang 48: Zweites Beispiel,
Berechnung nach dem Verfahren von Vetter

Vierte Gewichtung:

i \ j	1	2	3	4	5	6	7	8	9
1	0	0	0	0	0	0	0	0	0
2	3	0	0	0	0	0	0	0	0
3	5	4	0	0	0	2	0	0	0
4	C	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0
6	1	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0
9	C	0	0	0	0	0	0	0	0

Zwischenergebnis, bei dem inverse Abbildungen noch nicht zusammengefaßt wurden:

i \ j	1	2	3	4	5	6	7	8	9
1	0	0	0	0	0	0	0	0	0
2	1	0	0	0	0	1	0	0	0
3	0	1	0	0	1	0	0	0	0
4	C	0	1	0	0	0	1	1	0
5	0	0	0	0	0	0	0	0	0
6	0	1	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	1
8	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	1	0	0

Ergebnis, wenn inverse Abbildungen zusammengefaßt sind:

i \ j	1	2	3	4	5	6	7	8	9
1	0	0	0	0	0	0	0	0	0
2	1	1	0	0	0	1	0	0	0
3	C	1	0	0	1	0	0	0	0
4	0	0	1	0	0	0	1	1	0
5	0	0	0	0	0	0	0	0	0
6	C	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	1	0	1
8	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0

Anhang 49: Zweites Beispiel,
Berechnung nach dem Verfahren von Vetter

Fünfte Gewichtung:

i \ j	1	2	3	4	5	6	7	8	9
1	0	0	0	0	0	0	0	0	0
2	1	0	0	0	0	0	0	0	0
3	5	4	0	0	0	2	0	0	0
4	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0
6	3	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0

Zwischenergebnis, bei dem inverse Abbildungen noch nicht zusammengefaßt wurden:

i \ j	1	2	3	4	5	6	7	8	9
1	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	1	0	0	0
3	1	1	0	0	1	0	0	0	0
4	0	0	1	0	0	0	1	1	0
5	0	0	0	0	0	0	0	0	0
6	1	1	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	1
8	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	1	0	0

Ergebnis, wenn inverse Abbildungen zusammengefaßt sind:

i \ j	1	2	3	4	5	6	7	8	9
1	0	0	0	0	0	0	0	0	0
2	1	1	0	0	0	1	0	0	0
3	1	1	0	0	1	0	0	0	0
4	0	0	1	0	0	0	1	1	0
5	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	1	0	1
8	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0

Anhang 50: Zweites Beispiel,
Berechnung nach dem Verfahren von Vetter

Adjazenzmatrix für das dritte Beispiel:

i \ j	1	2	3	4	5	6
1	0	0	1	1	1	1
2	0	0	0	1	1	1
3	0	0	0	0	1	1
4	0	0	0	0	1	1
5	0	0	0	0	0	1
6	0	0	0	0	0	0

Die Relationen wurden in der Reihenfolge der maximalen Distanzen numeriert:

i \ j	1	2	3	4	5	6
1	0	0	7	8	2	1
2	0	0	0	9	4	3
3	0	0	0	0	10	5
4	0	0	0	0	11	6
5	0	0	0	0	0	12
6	0	0	0	0	0	0

Als Matrix der äquivalenten Relationen erhält man:

i \ j	1	2	3	4	5	6	7	8	9	10	11	12
1	1	4	0	0	2	3	2	3	0	0	0	4
2	0	1	0	0	0	0	2	3	0	2	3	0
3	0	0	1	3	0	2	0	0	2	0	0	3
4	0	0	0	1	0	0	0	0	2	0	2	0
5	0	0	0	0	1	0	0	0	0	2	0	2
6	0	0	0	0	0	1	0	0	0	0	2	2
7	0	0	0	0	0	0	1	0	0	0	0	0
8	0	0	0	0	0	0	0	1	0	0	0	0
9	0	0	0	0	0	0	0	0	1	0	0	0
10	0	0	0	0	0	0	0	0	0	1	0	0
11	0	0	0	0	0	0	0	0	0	0	1	0
12	0	0	0	0	0	0	0	0	0	0	0	1

Anhang 51: Drittes Beispiel,
Berechnung nach dem Verfahren von Wang und Wedekind

Für das dritte Beispiel erhält man als Attribute, die direkt und indirekt voneinander abhängig sind:

i \ j	1	2	3	4	5	6
1	0	0	0	0	1	1
2	0	0	0	0	1	1
3	0	0	0	0	0	1
4	0	0	0	0	0	1
5	0	0	0	0	0	0
6	0	0	0	0	0	0

es wurden folgende Gewichtungen gewählt:
erste Gewichtung:

i \ j	1	2	3	4	5	6
1	0	0	0	0	6	2
2	0	0	0	0	3	4
3	0	0	0	0	0	5
4	0	0	0	0	0	1
5	0	0	0	0	0	0
6	0	0	0	0	0	0

zweite Gewichtung:

i \ j	1	2	3	4	5	6
1	0	0	0	0	1	2
2	0	0	0	0	3	4
3	0	0	0	0	0	5
4	0	0	0	0	0	6
5	0	0	0	0	0	0
6	0	0	0	0	0	0

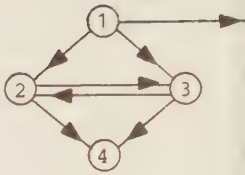
Ergebnis:

i \ j	1	2	3	4	5	6
1	0	0	1	1	0	0
2	0	0	0	1	0	0
3	0	0	0	0	1	0
4	0	0	0	0	1	0
5	0	0	0	0	0	1
6	0	0	0	0	0	0

Anhang 52: Drittes Beispiel,
Berechnung nach dem Verfahren von Vetter

MATRIX M1 VOM 82/01/13 14.26.18

I/J	1	2	3	4	5
1	0	1	1	1	1
2	0	0	1	1	0
3	0	1	0	1	0
4	0	0	0	0	0
5	0	0	0	0	0

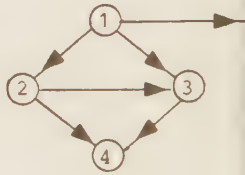


SCHLEIFENSUCHE VOM 82/01/13 14.26.18

SCHLEIFE IN KNOTEN
2 3

GRAPH OHNE SCHLEIFE VOM 82/01/13 14.26.18

I/J	1	2	3	4	5
1	0	1	1	1	1
2	0	0	1	1	0
3	0	0	0	1	0
4	0	0	0	0	0
5	0	0	0	0	0

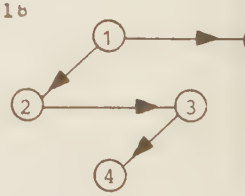


SUMME MAX. DISTANZ = 7
MAXIMALE LAENGE VON DER WURZEL ZU DEN KNOTEN

KNOTEN	1	2	3	4	5
DISTANZ	0	1	2	3	1

BAUM = MATRIX T1 VOM 82/01/13 14.26.18

I/J	1	2	3	4	5
1	0	1	0	0	1
2	0	0	1	0	0
3	0	0	0	1	0
4	0	0	0	0	0
5	0	0	0	0	0



ERREICHBARKEITSMATRIX VOM 82/01/13 14.26.18

I/J	1	2	3	4	5
1	1	1	1	1	1
2	0	1	1	1	0
3	0	0	1	1	0
4	0	0	0	1	0
5	0	0	0	0	1

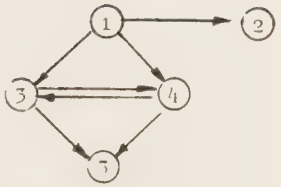
MAX ANZAHL DURCHLAUFENER KANTEN= 3 ODER 4
ERREICHBARKEIT FUER ALLE KNOTEN GEGEBEN
= ERREICHBARKEIT DES ZYKLENFREIEN GRAPHEN

NICHT ERREICHBAR FUER KNOTEN I= 3 J= 2
= ERREICHBARKEIT DES GRAPHEN MIT ZYKLEN

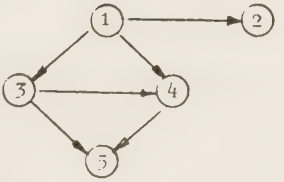
Relationen in
R1 (1,2,5)
R2 (2,3)
R3 (3,4)

Anhang 53: Ergebnis des ersten Beispiels

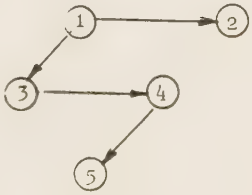
```
EINGABEPROTOKOLL
ANZSP=  5 ANZZL=  5
```




```
MATRIX M1              VOM      81/11/08 10.54.37

 I/J I   1   2   3   4   5
-----I-------------------------------
   1 I   0   1   1   1   1
   2 I   0   0   0   0   0
   3 I   0   0   0   1   1
   4 I   0   0   1   0   1
   5 I   0   0   0   0   0
SCHLEIFENSUCHE VOM     81/11/08 10.54.37
 SCHLEIFE IN   KNOTEN
   3   4
GRAPH OHNE SCHLEIFE    VOM 81/11/08 10.54.37
 I/J I   1   2   3   4   5
-----I-------------------------------
   1 I   0   1   1   1   1
   2 I   0   0   0   0   0
   3 I   0   0   0   1   1
   4 I   0   0   0   0   1
   5 I   0   0   0   0   0
```




```
MAXIMALE LAENGE VON DER WURZEL ZU DEN KNOTEN
KNOTEN      1   2   3   4   5
DISTANZ     0   1   1   2   3

BAUM = MATRIX T1           VOM      81/11/08 10.54.37

 I/J I   1   2   3   4   5
-----I-------------------------------
   1 I   0   1   1   0   0
   2 I   0   0   0   0   0
   3 I   0   0   0   1   0
   4 I   0   0   0   0   1
   5 I   0   0   0   0   0
```




```
ERREICHBARKEITSMATRIX      VOM      81/11/08 10.54.37

 I/J I   1   2   3   4   5
-----I-------------------------------
   1 I   1   1   1   1   1
   2 I   0   1   0   0   0
   3 I   0   0   1   1   1
   4 I   0   0   0   1   1
   5 I   0   0   0   0   1

MAX ANZAHL DURCHLAUFENER KANTEN=   3 ODER     4
ERREICHBARKEIT FUER ALLE KNOTEN GEGEBEN
= ERREICHBARKEIT DES ZYKLENFREIEN GRAPHEN

NICHT ERREICHBAR FUER KNOTEN I=   4 J=   1
= ERREICHBARKEIT DES GRAPHEN MIT ZYKLEN
```

Realtionen in 3NF

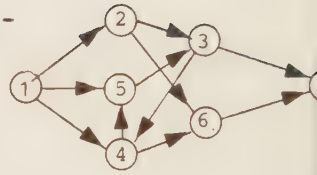
R1 (1,2,3)

R2 (3,4)

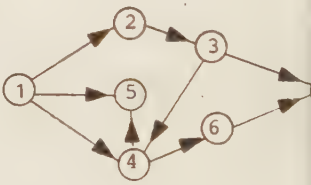
R3 (4,5)

Anhang 54: Ergebnisse des ersten Beispiels mit anderer Knotennumerierung

```
EINGABEPROTOKOLL
ANZSP=  7 ANZZL=  7


MATRIX M1          VOM      82/01/13 14.40.21

I/J I   1   2   3   4   5   6   7
---I-------------------------------
  1 I   0   1   0   1   1   0   0
  2 I   0   0   1   0   0   1   0
  3 I   0   0   0   1   0   0   1
  4 I   0   0   0   0   1   1   0
  5 I   0   0   1   0   0   0   0
  6 I   0   0   0   0   0   0   1
  7 I   0   0   0   0   0   0   0
```

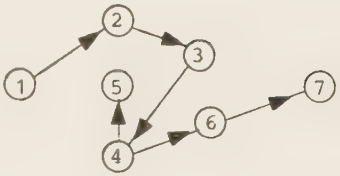



```
SCHLEIFENSUCHE VOM     82/01/13 14.40.21

SCHLEIFE IN   KNOTEN
  3   4   5


GRAPH OHNE SCHLEIFE    VOM 82/01/13 14.40.21
I/J I   1   2   3   4   5   6   7
---I-------------------------------
  1 I   0   1   0   1   1   0   0
  2 I   0   0   1   0   0   1   0
  3 I   0   0   0   1   0   0   1
  4 I   0   0   0   0   1   1   0
  5 I   0   0   0   0   0   0   0
  6 I   0   0   0   0   0   0   1
  7 I   0   0   0   0   0   0   0
```



Anhang 55: Zweites Beispiel

```
SUMME MAX. DISTANZ =   19
MAXIMALE LAENGE VON DER WURZEL ZU DEN KNOTEN
KNOTEN      1   2   3   4   5   6   7
DISTANZ     0   1   2   3   4   4   5


BAUM = MATRIX T1              VOM       82/01/13 14.40.21

I/J I   1   2   3   4   5   6   7
----I-------------------------------------
  1 I   0   1   0   0   0   0   0
  2 I   0   0   1   0   0   0   0
  3 I   0   0   0   1   0   0   0
  4 I   0   0   0   0   1   1   0
  5 I   0   0   0   0   0   0   1
  6 I   0   0   0   0   0   0   1
  7 I   0   0   0   0   0   0   0
```




```
ERREICHBARKEITSMATRIX         VOM       82/01/13 14.40.21

I/J I   1   2   3   4   5   6   7
----I-------------------------------------
  1 I   1   1   1   1   1   1   1
  2 I   0   1   1   1   1   1   1
  3 I   0   0   1   1   1   1   1
  4 I   0   0   0   1   1   1   1
  5 I   0   0   0   0   1   0   0
  6 I   0   0   0   0   0   1   1
  7 I   0   0   0   0   0   0   1

MAX ANZAHL DURCHLAUFENER KANTEN=   4 ODER    5
ERREICHBARKEIT FUER ALLE KNOTEN GEGEBEN
= ERREICHBARKEIT DES ZYKLENFREIEN GRAPHEN

NICHT ERREICHBAR FUER KNOTEN I=   5 J=   3
= ERREICHBARKEIT DES GRAPHEN MIT ZYKLEN
```

Relationen in 3NF:

R1 (1,2)

R2 (2,3)

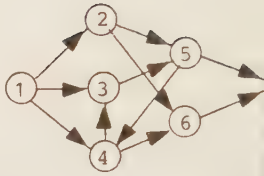
R3 (3,4)

R4 (4,5,6)

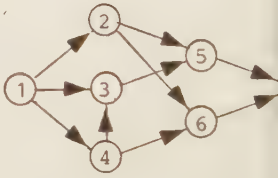
R5 (6,7)

Anhang 56: Ergebnis des zweiten Beispiels

```
EINGABEPROTOKOLL
ANZSP=  7 ANZZL=  7


MATRIX M1          VOM      82/01/13 14.33.35

I/J I   1   2   3   4   5   6   7
---I-------------------------------
  1 I   0   1   1   1   0   0   0
  2 I   0   0   0   0   1   1   0
  3 I   0   0   0   0   1   0   0
  4 I   0   0   1   0   0   1   0
  5 I   0   0   0   1   0   0   1
  6 I   0   0   0   0   0   0   1
  7 I   0   0   0   0   0   0   0
```

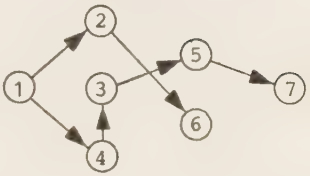



```
SCHLEIFENSUCHE VOM    82/01/13 14.33.35

SCHLEIFE IN   KNOTEN
  3   4   5


GRAPH OHNE SCHLEIFE    VOM 82/01/13 14.33.35
I/J I   1   2   3   4   5   6   7
---I-------------------------------
  1 I   0   1   1   1   0   0   0
  2 I   0   0   0   0   1   1   0
  3 I   0   0   0   0   1   0   0
  4 I   0   0   1   0   0   1   0
  5 I   0   0   0   0   0   0   1
  6 I   0   0   0   0   0   0   1
  7 I   0   0   0   0   0   0   0
```



Anhang 57: Zweites Beispiel mit geänderter Knotennumerierung

```
SUMME MAX. DISTANZ =   13
MAXIMALE LAENGE VON DER WURZEL ZU DEN KNOTEN
KNOTEN      1   2   3   4   5   6   7
DISTANZ     0   1   2   1   3   2   4


BAUM = MATRIX T1              VOM       82/01/13 14.33.35

I/J I   1   2   3   4   5   6   7
----I-------------------------------
  1 I   0   1   0   1   0   0   0
  2 I   0   0   0   0   0   1   0
  3 I   0   0   0   0   1   0   0
  4 I   0   0   1   0   0   0   0
  5 I   0   0   0   0   0   0   1
  6 I   0   0   0   0   0   0   0
  7 I   0   0   0   0   0   0   0
```




```
ERREICHBARKEITSMATRIX         VOM       82/01/13 14.33.35

I/J I   1   2   3   4   5   6   7
----I-------------------------------
  1 I   1   1   1   1   1   1   1
  2 I   0   1   0   0   0   1   0
  3 I   0   0   1   0   1   0   1
  4 I   0   0   1   1   1   0   1
  5 I   0   0   0   0   1   0   1
  6 I   0   0   0   0   0   1   0
  7 I   0   0   0   0   0   0   1

MAX ANZAHL DURCHLAUFENER KANTEN=   3 ODER    4
NICHT ERREICHBAR FUER KNOTEN I=   2 J=   5
NICHT ERREICHBAR FUER KNOTEN I=   4 J=   6
NICHT ERREICHBAR FUER KNOTEN I=   6 J=   7
= ERREICHBARKEIT DES ZYKLENFREIEN GRAPHEN

NICHT ERREICHBAR FUER KNOTEN I=   2 J=   5
NICHT ERREICHBAR FUER KNOTEN I=   4 J=   6
NICHT ERREICHBAR FUER KNOTEN I=   5 J=   4
NICHT ERREICHBAR FUER KNOTEN I=   6 J=   7
= ERREICHBARKEIT DES GRAPHEN MIT ZYKLEN
```

Relationen in 3NF:

R1 (1,2,4)

R2 (2,6)

R3 (3,5)

R4 (4,3)

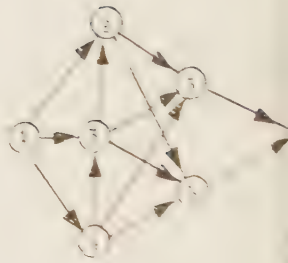
R5 (5,7)

Anhang 58: Ergebnis des zweiten Beispiels mit geänderter Knotennumerierung

EINGABEPROTOKOLL
ANZSP= 7 ANZEL= 7

MATRIX M1 VOM 81/11/12 11.50.07

I/J	1	2	3	4	5	6	7
1	0	1	1	1	0	0	0
2	0	0	0	0	1	1	0
3	0	1	0	0	1	1	0
4	0	0	1	0	1	1	0
5	0	0	0	0	0	0	1
6	0	0	0	0	0	0	1
7	0	0	0	0	0	0	0



SCHLEIFENSUCHE VOM 81/11/12 11.50.07

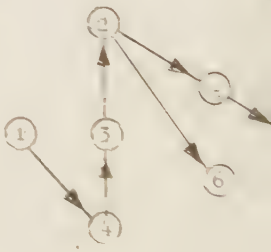
GRAPH HAT KEINE SCHLEIFEN

MAXIMALE LAENGE VON DER WURZEL ZU DEN KNOTEN

KNOTEN	1	2	3	4	5	6	7
DISTANZ	0	3	2	1	4	4	5

BAUM = MATRIX T1 VOM 81/11/12 11.50.07

I/J	1	2	3	4	5	6	7
1	0	0	0	1	0	0	0
2	0	0	0	0	1	1	0
3	0	1	0	0	0	0	0
4	0	0	1	0	0	0	0
5	0	0	0	0	0	0	1
6	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0



ERREICHBARKEITSMATRIX VOM 81/11/12 11.50.07

I/J	1	2	3	4	5	6	7
1	1	1	1	1	1	1	1
2	0	1	0	0	1	1	1
3	0	1	1	0	1	1	1
4	0	1	1	1	1	1	1
5	0	0	0	0	1	0	1
6	0	0	0	0	0	1	0
7	0	0	0	0	0	0	1

MAX ANZAHL DURCHLAUFENER KANTEN= 4 ODER 5
NICHT ERREICHBAR FUER KNOTEN I= 6 J= 7
= ERREICHBARKEIT DES ZYKLENFREIEN GRAPHEN

Relationen in 3NF

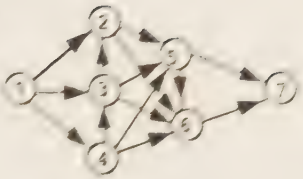
R1 (1,4)
R2 (2,5,6)
R3 (3,2)
R4 (4,3)
R5 (5,7)

Anhang 5: Ergebnisse des zweiten abgewandelten Beispiels

EINGABEPROTOKOLL
ANZSP= 7 ANZZL= 7

MATRIX M1 VOM 82/01/13 14.44.43

I/J	1	2	3	4	5	6	7
1	0	1	1	1	0	0	0
2	0	0	0	0	1	1	0
3	0	1	0	0	1	1	0
4	0	0	1	0	1	1	0
5	0	0	0	0	0	1	1
6	0	0	0	0	0	0	1
7	0	0	0	0	0	0	0



SCHLEIFENSUCHE VOM 82/01/13 14.44.43

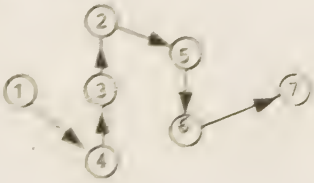
GRAPH HAT KEINE SCHLEIFEN

SUMME MAX. DISTANZ = 21
MAXIMALE LAENGE VON DER WURZEL ZU DEN KNOTEN

KNOTEN	1	2	3	4	5	6	7
DISTANZ	0	3	2	1	4	5	6

BAUM = MATRIX T1 VOM 82/01/13 14.44.43

I/J	1	2	3	4	5	6	7
1	0	0	0	1	0	0	0
2	0	0	0	0	1	0	0
3	0	1	0	0	0	0	0
4	0	0	1	0	0	0	0
5	0	0	0	0	0	1	0
6	0	0	0	0	0	0	1
7	0	0	0	0	0	0	0



ERREICHBARKEITSMATRIX VOM 82/01/13 14.44.43

I/J	1	2	3	4	5	6	7
1	1	1	1	1	1	1	1
2	0	1	0	0	1	1	1
3	0	1	1	0	1	1	1
4	0	1	1	1	1	1	1
5	0	0	0	0	1	1	1
6	0	0	0	0	0	1	1
7	0	0	0	0	0	0	1

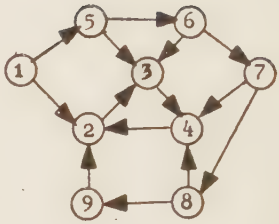
Relationen in 3NF:
R1 (1,4)
R2 (2,5)
R3 (3,2)
R4 (4,3)
R5 (5,6)
R6 (6,7)

MAX ANZAHL DURCHLAUFENER KANTEN= 4 ZWEG 5
ERREICHBARKEIT FUER ALLE KNOTEN BEGINNT
= ERREICHBARKEIT DES ZYKLENFREIEN GRAPHEN

Anlage 61: Ergebnis des eingewandelten zweiten Beispiels

.MATRIX M1 VOM 81/12/17 13.51.15

I/J	1	2	3	4	5	6	7	8	9
1	0	1	0	0	1	0	0	0	0
2	0	0	1	0	0	0	0	0	0
3	0	0	0	1	0	0	0	0	0
4	0	1	0	0	0	0	0	0	0
5	0	0	1	0	0	1	0	0	0
6	0	0	1	0	0	0	1	0	0
7	0	0	0	1	0	0	0	1	0
8	0	0	0	1	0	0	0	0	1
9	0	1	0	0	0	0	0	0	0

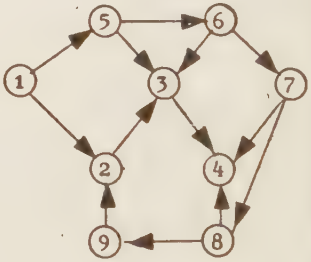


.SCHLEIFENSUCHE VOM 81/12/17 13.51.15

SCHLEIFE IN KNOTEN
2 3 4

GRAPH OHNE SCHLEIFE VOM 81/12/17 13.51.15

I/J	1	2	3	4	5	6	7	8	9
1	0	1	0	0	1	0	0	0	0
2	0	0	1	0	0	0	0	0	0
3	0	0	0	1	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0
5	0	0	1	0	0	1	0	0	0
6	0	0	1	0	0	0	1	0	0
7	0	0	0	1	0	0	0	1	0
8	0	0	0	1	0	0	0	0	1
9	0	1	0	0	0	0	0	0	0

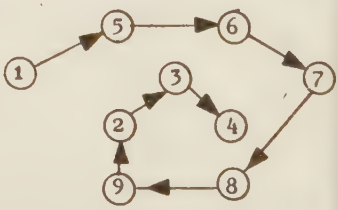


SUMME MAX. DISTANZ = 36
MAXIMALE LAENGE VON DER WURZEL ZU DEN KNOTEN

KNOTEN	1	2	3	4	5	6	7	8	9
DISTANZ	0	6	7	8	1	2	3	4	5

.BAUM = MATRIX T1 VOM 81/12/17 13.51.15

I/J	1	2	3	4	5	6	7	8	9
1	0	0	0	0	1	0	0	0	0
2	0	0	1	0	0	0	0	0	0
3	0	0	0	1	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	1	0	0
7	0	0	0	0	0	0	0	1	0
8	0	0	0	0	0	0	0	0	1
9	0	1	0	0	0	0	0	0	0



Anhang 61: Ergebnisse des dritten Beispiels

ERREICHBARKEITSMATRIX VOM 81/12/17 13.51.15

I/J	I	1	2	3	4	5	6	7	8	9
1	I	1	1	1	1	1	1	1	1	1
2	I	0	1	1	1	0	0	0	0	0
3	I	0	0	1	1	0	0	0	0	0
4	I	0	0	0	1	0	0	0	0	0
5	I	0	1	1	1	1	1	1	1	1
6	I	0	1	1	1	0	1	1	1	1
7	I	0	1	1	1	0	0	1	1	1
8	I	0	1	1	1	0	0	0	1	1
9	I	0	1	1	1	0	0	0	0	1

MAX ANZAHL DURCHLAUFENER KANTEN= 4 ODER 5
ERREICHBARKEIT FUER ALLE KNOTEN GEGEBEN
= ERREICHBARKEIT DES ZYKLENFREIEN GRAPHEN

NICHT ERREICHBAR FUER KNOTEN I= 4 J= 2
= ERREICHBARKEIT DES GRAPHEN MIT ZYKLEN

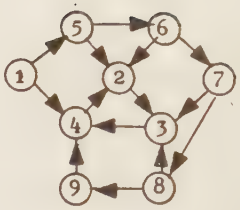
Relationen in 3NF

R1 (1,5)
R2 (2,3)
R3 (3,4)
R4 (5,6)
R5 (6,7)
R6 (7,8)
R7 (8,9)
R8 (9,2)

Anhang 62: Ergebnisse des dritten Beispiels

MATRIX M1 VOM 81/12/17 13.52.37

I/J	1	2	3	4	5	6	7	8	9
1	0	0	0	1	1	0	0	0	0
2	0	0	1	0	0	0	0	0	0
3	0	0	0	1	0	0	0	0	0
4	0	1	0	0	0	0	0	0	0
5	0	1	0	0	0	1	0	0	0
6	0	1	0	0	0	0	1	0	0
7	0	0	1	0	0	0	0	1	0
8	0	0	1	0	0	0	0	0	1
9	0	0	0	1	0	0	0	0	0

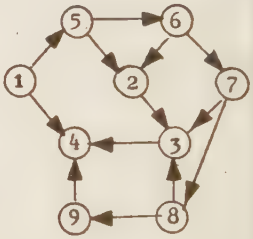


SCHLEIFENSUCHE VOM 81/12/17 13.52.37

SCHLEIFE IN KNOTEN
2 3 4

GRAPH OHNE SCHLEIFE VOM 81/12/17 13.52.37

I/J	1	2	3	4	5	6	7	8	9
1	0	0	0	1	1	0	0	0	0
2	0	0	1	0	0	0	0	0	0
3	0	0	0	1	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0
5	0	1	0	0	0	1	0	0	0
6	0	1	0	0	0	0	1	0	0
7	0	0	1	0	0	0	0	1	0
8	0	0	1	0	0	0	0	0	1
9	0	0	0	1	0	0	0	0	0

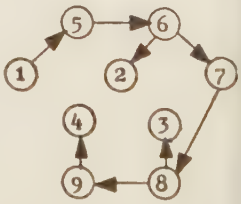


SUMME MAX. DISTANZ = 29
MAXIMALE LAENGE VON DER WURZEL ZU DEN KNOTEN

KNOTEN	1	2	3	4	5	6	7	8	9
DISTANZ	0	3	5	6	1	2	3	4	5

BAUM = MATRIX T1 VOM 81/12/17 13.52.37

I/J	1	2	3	4	5	6	7	8	9
1	0	0	0	0	1	0	0	0	0
2	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	1	0	0	0
6	0	1	0	0	0	0	1	0	0
7	0	0	0	0	0	0	0	1	0
8	0	0	1	0	0	0	0	0	1
9	0	0	0	1	0	0	0	0	0



Anhang 63 : Ergebnisse des dritten Beispiels mit anderer Knotennumerierung

ERREICHBARKEITSMATRIX VOM 81/12/17 13.52.37

I/J	1	2	3	4	5	6	7	8	9
1	1	1	1	1	1	1	1	1	1
2	0	1	0	0	0	0	0	0	0
3	0	0	1	0	0	0	0	0	0
4	0	0	0	1	0	0	0	0	0
5	0	1	1	1	1	1	1	1	1
6	0	1	1	1	0	1	1	1	1
7	0	0	1	1	0	0	1	1	1
8	0	0	1	1	0	0	0	1	1
9	0	0	0	1	0	0	0	0	1

Relationen in 3NF

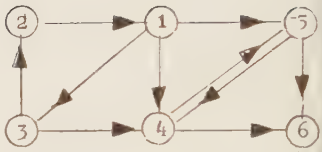
R1 (1,5)
R2 (5,6)
R3 (6,2,7)
R4 (7,8)
R5 (8,3,9)
R6 (9,4)

MAX ANZAHL DURCHLAUFENER KANTEN= 4 ODER 5
NICHT ERREICHBAR FUER KNOTEN I= 2 J= 3
NICHT ERREICHBAR FUER KNOTEN I= 3 J= 4
- ERREICHBARKEIT DES ZYKLENFREIEN GRAPHEN

NICHT ERREICHBAR FUER KNOTEN I= 2 J= 3
NICHT ERREICHBAR FUER KNOTEN I= 3 J= 4
NICHT ERREICHBAR FUER KNOTEN I= 4 J= 2
- ERREICHBARKEIT DES GRAPHEN MIT ZYKLEN

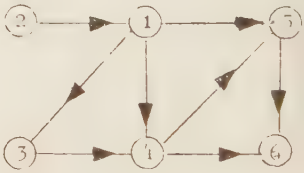
Anhang 64: Ergebnisse des dritten Beispiels mit anderer Knotennumerierung

```
EINGABEPROTOKOLL
ANZSP=  6 ANZZL=  6
```




```
MATRIX M1              VOM      81/11/12 12.06.23

 I/J I   1   2   3   4   5   6
-----I-------------------------------
   1 I   0   0   1   1   1   0
   2 I   1   0   0   0   0   0
   3 I   0   1   0   1   0   0
   4 I   0   0   0   0   1   1
   5 I   0   0   0   1   0   1
   6 I   0   0   0   0   0   0
```

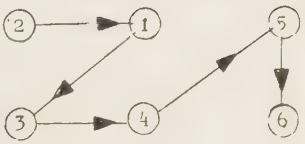
```
SCHLEIFENSUCHE VOM     81/11/12 12.06.23

SCHLEIFE IN    KNOTEN
  1   2   3

SCHLEIFE IN    KNOTEN
  4   5
```




```
GRAPH OHNE SCHLEIFE    VOM 81/11/12 12.06.23
 I/J I   1   2   3   4   5   6
-----I-------------------------------
   1 I   0   0   1   1   1   0
   2 I   1   0   0   0   0   0
   3 I   0   0   0   1   0   0
   4 I   0   0   0   0   1   1
   5 I   0   0   0   0   0   1
   6 I   0   0   0   0   0   0
```

Anhang 65: Viertes Beispiel

```
AXIMALE LAENGE VON DER WURZEL ZU DEN KNOTEN
NOTEN      1   2   3   4   5   6
ISTANZ     1   0   2   3   4   5


AUM = MATRIX T1              VOM       81/11/12 12.06.23

/J I   1   2   3   4   5   6
---I-------------------------------------
 1 I   0   0   1   0   0   0
 2 I   1   0   0   0   0   0
 3 I   0   0   0   1   0   0
 4 I   0   0   0   0   1   0
 5 I   0   0   0   0   0   1
 6 I   0   0   0   0   0   0
```



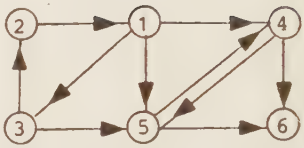

```
RREICHBARKEITSMATRIX          VOM       81/11/12 12.06.23

/J I   1   2   3   4   5   6
---I-------------------------------------
 1 I   1   0   1   1   1   1
 2 I   1   1   1   1   1   1
 3 I   0   0   1   1   1   1
 4 I   0   0   0   1   1   1
 5 I   0   0   0   0   1   1
 6 I   0   0   0   0   0   1

AX ANZAHL DURCHLAUFENER KANTEN=   4 ODER     5
RREICHBARKEIT FUER ALLE KNOTEN GEGEBEN
 ERREICHBARKEIT DES ZYKLENFREIEN GRAPHEN

ICHT ERREICHBAR FUER KNOTEN I=   3 J=   2
ICHT ERREICHBAR FUER KNOTEN I=   5 J=   4
 ERREICHBARKEIT DES GRAPHEN MIT ZYKLEN
```

Relationen in 3NF:

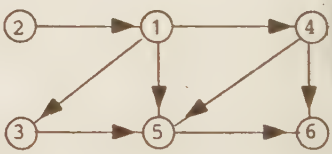
R1 (1,3)
R2 (2,1)
R3 (3,4)
R4 (4,5)
R5 (5,6)

Anhang 66: Ergebnisse des vierten Beispiels

```
EINGABEPROTOKOLL
ANZSP=  6 ANZZL=  6


MATRIX M1          VOM     82/01/13 15.19.57

I/J I   1   2   3   4   5   6
---I-------------------------------
  1 I   0   0   1   1   1   0
  2 I   1   0   0   0   0   0
  3 I   0   1   0   0   1   0
  4 I   0   0   0   0   1   1
  5 I   0   0   0   1   0   1
  6 I   0   0   0   0   0   0
```

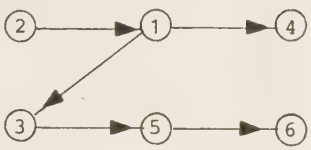



```
SCHLEIFENSUCHE VOM     82/01/13 15.19.57

SCHLEIFE IN   KNOTEN
  1   2   3

SCHLEIFE IN   KNOTEN
  4   5


GRAPH OHNE SCHLEIFE    VOM 82/01/13 15.19.57
I/J I   1   2   3   4   5   6
---I-------------------------------
  1 I   0   0   1   1   1   0
  2 I   1   0   0   0   0   0
  3 I   0   0   0   0   1   0
  4 I   0   0   0   0   1   1
  5 I   0   0   0   0   0   1
  6 I   0   0   0   0   0   0
```



Anhang 67: Viertes Beispiel mit geänderter Knotennumerierung

```
SUMME MAX. DISTANZ =   12
MAXIMALE LAENGE VON DER WURZEL ZU DEN KNOTEN
KNOTEN      1   2   3   4   5   6
DISTANZ     1   0   2   2   3   4


BAUM = MATRIX T1                VOM       82/01/13 15.19.57

I/J I   1   2   3   4   5   6
----I-------------------------------------
  1 I   0   0   1   1   0   0
  2 I   1   0   0   0   0   0
  3 I   0   0   0   0   1   0
  4 I   0   0   0   0   0   0
  5 I   0   0   0   0   0   1
  6 I   0   0   0   0   0   0
```




```
ERREICHBARKEITSMATRIX           VOM       82/01/13 15.19.57

I/J I   1   2   3   4   5   6
----I-------------------------------------
  1 I   1   0   1   1   1   1
  2 I   1   1   1   1   1   1
  3 I   0   0   1   0   1   1
  4 I   0   0   0   1   0   0
  5 I   0   0   0   0   1   1
  6 I   0   0   0   0   0   1

MAX ANZAHL DURCHLAUFENER KANTEN=   3 ODER    4
NICHT ERREICHBAR FUER KNOTEN I=   4 J=   5
NICHT ERREICHBAR FUER KNOTEN I=   4 J=   6
= ERREICHBARKEIT DES ZYKLENFREIEN GRAPHEN

NICHT ERREICHBAR FUER KNOTEN I=   3 J=   2
NICHT ERREICHBAR FUER KNOTEN I=   4 J=   5
NICHT ERREICHBAR FUER KNOTEN I=   4 J=   6
NICHT ERREICHBAR FUER KNOTEN I=   5 J=   4
= ERREICHBARKEIT DES GRAPHEN MIT ZYKLEN
```

Relationen in 3NF:

R1 (1,3,4)

R2 (2,1)

R3 (3,5)

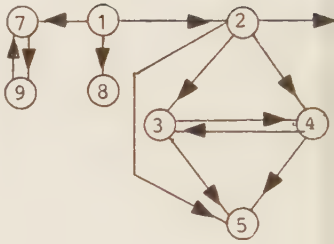
R4 (5,6)

Anhang 68: Ergebnis des vierten Beispiels mit geänderter Knotennumerierung

EINGABE KOMPLETT
ANZSP= 9 ANZZL= 9

MATRIX M1 VOM 82/01/13 14.46.35

I/J	1	2	3	4	5	6	7	8	9
1	0	1	0	0	0	0	1	1	0
2	0	0	1	1	1	1	0	0	0
3	0	0	0	1	1	0	0	0	0
4	0	0	1	0	1	0	0	0	0
5	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	1
8	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	1	0	0



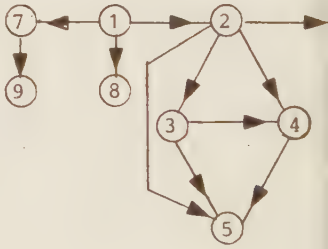
SCHLEIFENSUCHE VOM 82/01/13 14.46.35

SCHLEIFE IN KNOTEN
3 4

SCHLEIFE IN KNOTEN
7 9

GRAPH OHNE SCHLEIFE VOM 82/01/13 14.46.35

I/J	1	2	3	4	5	6	7	8	9
1	0	1	0	0	0	0	1	1	0
2	0	0	1	1	1	1	0	0	0
3	0	0	0	1	1	0	0	0	0
4	0	0	0	0	1	0	0	0	0
5	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	1
8	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0



Anhang 69: Fünftes Beispiel

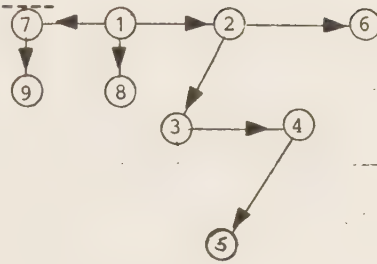
SUMME MAX. DISTANZ = 16

MAXIMALE LAENGE VON DER WURZEL ZU DEN KNOTEN

KNOTEN	1	2	3	4	5	6	7	8	9
DISTANZ	0	1	2	3	4	2	1	1	2

BAUM = MATRIX T1 VOM 82/01/13 14.46.35

I/J	1	2	3	4	5	6	7	8	9
1	0	1	0	0	0	0	1	1	0
2	0	0	1	0	0	1	0	0	0
3	0	0	0	1	0	0	0	0	0
4	0	0	0	0	1	0	0	0	0
5	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	1
8	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0



ERREICHBARKEITSMATRIX VOM 82/01/13 14.46.35

I/J	1	2	3	4	5	6	7	8	9
1	1	1	1	1	1	1	1	1	1
2	0	1	1	1	1	1	0	0	0
3	0	0	1	1	1	0	0	0	0
4	0	0	0	1	1	0	0	0	0
5	0	0	0	0	1	0	0	0	0
6	0	0	0	0	0	1	0	0	0
7	0	0	0	0	0	0	1	0	1
8	0	0	0	0	0	0	0	1	0
9	0	0	0	0	0	0	0	0	1

Relationen in 3NF:

R1 (1,2,7,8)

R2 (2,3,6)

R3 (3,4)

R4 (4,5)

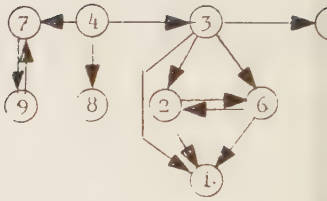
R5 (7,9)

MAX ANZAHL DURCHLAUFENER KANTEN= 3 ODER 4
ERREICHBARKEIT FUER ALLE KNOTEN GEGEBEN
= ERREICHBARKEIT DES ZYKLENFREIEN GRAPHEN

NICHT ERREICHBAR FUER KNOTEN I= 4 J= 3
NICHT ERREICHBAR FUER KNOTEN I= 9 J= 7
= ERREICHBARKEIT DES GRAPHEN MIT ZYKLEN

Anhang 70: Ergebnis des fünften Beispiels

.EINGABEPROTOKOLL
ANZSP= 9 ANZZL= 9



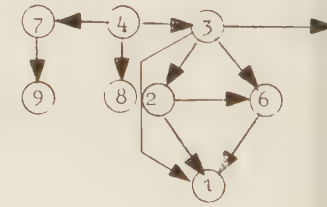
MATRIX M1 VOM 81/11/18 10.40.11

I/J	1	2	3	4	5	6	7	8	9
1	0	0	0	0	0	0	0	0	0
2	1	0	0	0	0	1	0	0	0
3	1	1	0	0	1	1	0	0	0
4	0	0	1	0	0	0	1	1	0
5	0	0	0	0	0	0	0	0	0
6	1	1	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	1
8	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	1	0	0

.SCHLEIFENSUCHE VOM 81/11/18 10.40.11

SCHLEIFE IN KNOTEN
2 6

SCHLEIFE IN KNOTEN
7 9



GRAPH OHNE SCHLEIFE VOM 81/11/18 10.40.11

I/J	1	2	3	4	5	6	7	8	9
1	0	0	0	0	0	0	0	0	0
2	1	0	0	0	0	1	0	0	0
3	1	1	0	0	1	1	0	0	0
4	0	0	1	0	0	0	1	1	0
5	0	0	0	0	0	0	0	0	0
6	1	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	1
8	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0

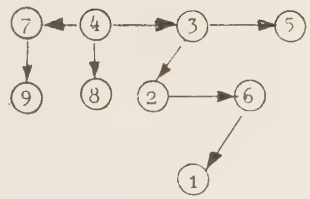
Anhang 71: Ergebnisse des fünften Beispiels
mit anderer Knotennumerierung

AXIMALE LAENGE VON DER WURZEL ZU DEN KNOTEN
UMME MAX. DISTANZ = 16

NOTEN	1	2	3	4	5	6	7	8	9
ISTANZ	4	2	1	0	2	3	1	1	2

AUM = MATRIX T1 VOM 81/11/18 10.40.11

VJ I	1	2	3	4	5	6	7	8	9
1 I	0	0	0	0	0	0	0	0	0
2 I	0	0	0	0	0	1	0	0	0
3 I	0	1	0	0	1	0	0	0	0
4 I	0	0	1	0	0	0	1	1	0
5 I	0	0	0	0	0	0	0	0	0
6 I	1	0	0	0	0	0	0	0	0
7 I	0	0	0	0	0	0	0	0	1
8 I	0	0	0	0	0	0	0	0	0
9 I	0	0	0	0	0	0	0	0	0



RREICHBARKEITSMATRIX VOM 81/11/18 10.40.11

VJ I	1	2	3	4	5	6	7	8	9
1 I	1	0	0	0	0	0	0	0	0
2 I	1	1	0	0	0	1	0	0	0
3 I	1	1	1	0	1	1	0	0	0
4 I	1	1	1	1	1	1	1	1	1
5 I	0	0	0	0	1	0	0	0	0
6 I	1	0	0	0	0	1	0	0	0
7 I	0	0	0	0	0	0	1	0	1
8 I	0	0	0	0	0	0	0	1	0
9 I	0	0	0	0	0	0	0	0	1

Relationen in 3NF:

R1 (2,6)
R2 (3,2,5)
R3 (4,3,7,8)
R4 (6,1)
R5 (7,9)

AX ANZAHL DURCHLAUFENER KANTEN= 3 ODER 4
RREICHBARKEIT FUER ALLE KNOTEN GEGEBEN
ERREICHBARKEIT DES ZYKLENFREIEN GRAPHEN

ICHT ERREICHBAR FUER KNOTEN I= 6 J= 2
ICHT ERREICHBAR FUER KNOTEN I= 9 J= 7
ERREICHBARKEIT DES GRAPHEN MIT ZYKLEN

Anhang 72 : Ergebnisse des fünften Beispiels mit anderer Knotennummerierung

Verfahren / Kriterien	Kapitel 2.1				2.2	2.3				Sonst.		3.3
	2.1.2 Maier, Freudenth.	2.1.3 Späth	2.1.5 Döringer	2.1.6 Koreimann	2.2 Smith, Smith	2.3.2 Wang Wedekind	2.3.3 Bernstein	2.3.4 Pagin	2.3.5 Vetter	Härder	Hau	3.3 neues Verfahren
Bildung von Relationen aufgrund												
- von Anwendungen	J	J	J	J								
- typischer Merkmale					J							
- funktionaler Abhängigkeiten						J	J	J	J			J
Berücksichtigung der Bedeutung von Attributen												
- durch Zugriffsart	N	N	J	N	N	N	N	N	N			N
- durch Reihenfolge als Ordnungskriterium				J								
- durch Zugriffshäufigkeit bei sonst gleichwertigen Lösungen												J
Definition der trans.Abhängigkeit												
- eine fkt. Abhängigkeit kann durch genau zwei andere ersetzt werden						J			J			J
- eine fkt. Abhängigkeit kann durch eine beliebige Kombination anderer ersetzt werden						-			A			J
Berücksichtigung von Zyklen												
- durch Zusammenfassung zu einem Hilfsknoten										J		
- durch Sonderbehandlung bei der Best. der min. Überdeckung											J	
- durch Auflösen, unter der Bedingung, daß die max. Erreichbarkeit erhalten wird												J
Normalisierung in optimale Normalform												
- mit min. Überdeckung							BCNF	4NF	-			
- mit min. Überdeckung auf der Basis der max.Distanz						3NF			3NF			
- durch sonst. Verfahren			3NF									
- mit Baumalgorithmus												3NF
Besonderheiten												
- Schlüssel kann aus Attributgruppen best.						A	J	-	A			A
- m:n Verb. wird durch Hilfsknoten aufgelöst						-	J	-	J			J
- multivalued dependencies werden berücksichtigt								J				
- inverse Abhängigkeiten ergeben 1 oder 2 Rel.						2	1	-	1			1
Überprüfung der Erreichbarkeit												
- aufgrund bekannter Anforderungen									J			
- aufgrund funktionaler Abhängigkeiten												J

Legende: J = ja
N = nein
A = ausgeschlossen
- = nicht explizit beschrieben
1,2 = Anzahl der gebildeten Relationen

LITERATURVERZEICHNIS

ADV-ORGA (Hrsg.): Seminarunterlagen Datenbankentwurfstechniken. ORGTECH 1. Wilhelmshaven 1980

Adiba, M., Delobel, C., Léonard, M: A Unified Approach for Modelling Data Base Design. In: Modelling in Data Base Management Systems, hrsg. von G.M.Nijssen. Amsterdam 1976, S. 311-338

ANSI/x3/SPARC: Study Group on Data Base Management Systems, Interim Report 75-02-08. In: FDT-Bulletin of ACM-SIGMOD, 7 (1975)

Bachman, C.W.: Data Structure Diagrams. Data Base, 1, 2 (1969), S. 4-10

Baldissera, C., Ceri, S., Pelagatti, G., Bracchi, G.: Interactive Specification and Formal Verification of User's Views in Database Design. In: Proc. 5th Int. Conf. on Very Large Data Bases. Rio de Janeiro 1979, S. 262-271

Beeri, C., Bernstein, P.A., Goodman, N.: A Sophisticate's Introduction to Database Normalization Theorey. In: Proc 3rd Int. Conf. on Very Large Data Bases, Tokyo 3.-8.Okt. 1977. New York 1977., S. 113-124

Berild, S.: CS4- A Tool for Database Design by Infological Simulation. In: Proc. 3rd Int. Conf. on Very Large Data Bases, Tokyo 3.-8. Okt. 1977. New York 1977, S. 533

Bernstein, P.A.: Normalization and Functional Dependencies in the Relational Data Base Model. Tech. Report CSRG-60, Ph. D. Diss., Computer Systems Research Group, Dep. Computer Science, University of Toronto, Toronto, Canada Okt. 1975

Bernstein, P.A.: Synthesizing Third Normal Form Relations from Functional Decpendencies. ACM Transactions on Data Base Systems, 1 (1976), S. 277-298

Bernstein, P.A.: Comment on "Segment Synthesis in Logical Data Base Design." IBM-Journal of Research and Development. 20, 4 (1976), S. 412

Bernstein, P.A, Beeri, C.: An Algorithmic Approach to Normalization of Relational Data Base Schemas. Tech. Report CSRG-73, Computer Systems Researc Group. Dept. Computer Science, University of Toronto, Toronto Canada, Sept. 1976

Bubenko jr., J.A., Berild, S., Lindencrona-Ohlin, E., Nachmens, S From Information Requirements to DBTG - Data Structures. In: ACM SIGPLAN/SIGMOD Conference, March 22-24, 1976, Salt Lake City, hrsg. von H. Ledgard und R. Taylor, Utah, USA

Bubenko jr. J.A.: IAM: An Inferential Abstract Modeling Approach to Design of Conceptual Schema. In: ACM SIGMOD, Int. Conf. on Management of Data, hrsg. von D.C.P. Smith. New York 1977, S. 62-74

Bubenko jr. J.A.: Data Models and their Semantics. In: INFOTECH State of the Art Report. Data Design, Invited Papers. Maidenhead 1980, S. 109-136

Chen, P.P.-S.: The Entity-Relationship Model - A Base for the Enterprise View of Data. In: AFIPS Conference Proc., 1977 National Computer Conference, hrsg. von R.R.Korfhage. Montvale, New Jersey 1977, S. 77-84

Chen, P.P.-S.: The Entity-Relationship Model - Toward a Unified View of Data. ACM Transaction on Data Base Systems, 1 (1976), S. 9-36

Chen, P.P.-S., Yao, S.B.: Design and Performance Tools for Data Base Systems. In: Proc. of the 3rd Int. Conf. on Very Large Data Bases. New York 1977, S.3-1

Codd, E.F.: Further Normalization of the Data Base Relational Model. In: Courant Computer Science Symposia 6: Data Base Systems, hrsg. von R. Rustin. Prentice Hall, Englewood Cliffs, N.J., 1972, S. 33-64

Codd, E.F.: A Relational Model of Data for Large Shared Data Bases. Communication of the ACM, 13 (1970), S. 377- 387

Davenport, B.A.: Data Analysis for Database Design. The Australian Computer Journal, 10 (1978), 4, S. 122-137

Davenport, R.A.: Data Analysis - Experience with a Formal Methodology. In: Proc. of the European Conf. on Applied Inf. Technology, hrsg. von P.A. Samet. Amsterdam 1979

Delobel, C.: A Theory About Data in an Information System. IBM Research Report RJ964, San José.Jan.1972

Delobel, C., Casey, R.G.: Decomposition of a Data Base and the Theory of Boolean Switching Functions. IBM Journal of Research and Development. 17 (1973), S. 374-386

Dörfler, W.: Mathematik für Informatiker. Band I. München 1977

Dörfler, W., Mühlbacher, J.: Graphentheorie für Informatiker. Berlin 1973

Döringer, W.: Eine Methodik zur Strukturierung von Datenbanken. Angewandte Informatik 1978, S. 93-103

Fadous, R., Forsyth, J.: Finding Candidate Keys for Relational Data Bases. In: Proc. of ACM-SIGMOD Int. Conf. on Management of Data, hrsg. von W.F. King. New York 1975, S. 203-210

Fagin, R.: Multivalued Dependencies and a New Normal Form for Relational Databases. IBM RJ1812 (26109), IBM Research Laboratory San José, Calif. 1976

Fagin, R.: Multivalued Dependencies and a New Normal Form for Relational Databases. ACM Transaction on Database Systems, 2 (1977), S. 262-278

Fagin, R.: The Decomposition Versus the Synthetic Approach to Relational Database Design. In: Proc. of the 3rd Int. Conf. on Very Large Databases. New York 1977, S. 441-446

Falkenberg, E., Hahn, H.: Information Analysis of Power Distribution Networks. Siemens Forschungs- u. Entwicklungsbericht Bd. 10 (1981), Berlin 1981, S. 246-253

Flory, A., Kouloumdjian, J.: A Model and a Method for Logical Data Base Design. In: Proc. of the Int. Conf. on Very Large Data Bases, o.O. 1978

Freudenthaler, B., Maier, B.: Die Entwicklung eines Datenmodells aus Informationsbedarfsanalysen - ein Klassifikationsproblem. Angewandte Informatik. 1977, S. 522-532

Grochla, E., Garbe, H., Gillner, R.: Gestaltungskriterien für den Aufbau von Datenbanken. Opladen 1973

Härder, Th.: Zur Synthese von Relationen in optimaler dritter Normalform. Forschungsbericht DVI 76-3, Forschungsgruppe Datenverwaltungssysteme. Darmstadt 1976

Härder, Th.: Implementierung von Datenbanksystemen. München 1978

Hanssmann, F.: Einführung in die Systemforschung. München 1978

Housel, B.C., Waddle, V., Yao, S.B.: The Functional Dependency Model for Logical Database Design. In: Proc. 5th Int. Conf. on Very Large Databases. Rio de Janeiro 1979, S. 194-203

HSU, H.T.: An Algorithm for Finding a Minimal Equivalent Graph of a Digraph. Journal of the ACM, 22 (1975), S. 11-16

Hu, Te Chiang: Ganzzahlige Programmierung und Netzwerkflüsse. München 1972

Hubbard, G.U.: Techniques for Automated Logical File Design. In: NYU Symposium of Database Design. New York 1978

Kahn, B.K.: A Structured Logical Database Design Methodology. In: NYU Symposium on Database Design. New York 1978

Koreimann, D.S.: Anwendungsanalyse - ein pragmatischer Ansatz für den Entwurf komplexer Datenbank-Organisationen. Angewandte Informatik, 1974, S.1-5

Krieger, R., Lausen, G., Stucky, W.: Methodischer Entwurf von Datenstrukturen für betriebliche Informationssysteme. (Kurzfassung zum Vortrag), o.O., o.J.

Krieger, R., Lausen, G., Stucky, W.: Methodischer Datenentwurf beim Aufbau betrieblicher Informationssysteme Angewandte Informatik, 1982, S. 191-198

Martens, J.-M.: Anwendungsbezogene Datenorganisation formatierter Datenbanken. München 1979

Meller, Ch.: Eine Methode des informationsbedarfsorientierten Datenbankentwurfs. Angewandte Informatik, 1980, S. 224-228

Mertens, P., Griese, J.: Datenbanken und Datenbanksysteme. In: Handwörterbuch der Betriebswirtschaftslehre Bd. 1/1, hrsg. von E.Grochla und W.Wittmann. Stuttgart 1974, Spalte 1088-1101

Meyer, M., Hansen, K., Rohde, M.: Mathematische Planungsverfahren I. Essen 1973

Mijares, T., Peebles, R.: A Methodology for the Design of Logical Data Base Structures. In: Modelling in Data Base Management Systemes, hrsg. von G.M. Nijssen. Amsterdam 1976, S. 25-47

Mitoma, M.F., Irani, K.B.: Automatic Data Base Schema Design and Optimization. Data Base Design Methodology, an Intensive Short Course, Engineering Summer Conf., Ann Arbor Michigan 1976, S. 286-321

Neumann, K.: Operations Research Verfahren Bd. 3: Graphentheorie-Netzplantechnik. München 1975

Nijssen, G.M.: A Framework for Advanced Mass Storage Applications. In: Medinfo 80, Proc. of 3rd World Conf. on Medical Informatics. Amsterdam 1980

Österle, H.: Entwurf betrieblicher Informationssysteme. München 1981

Österle, H.: Zum Entity-Relationship-Ansatz im Requirements Engineering. Angewandte Informatik, 1982, S. 413-420

Pape, U.: Kürzeste Wege in Netzwerken zwischen festen Knoten. In: Veröffentlichungen des Instituts für Stadtbauwesen der Technischen Universität Braunschweig. Heft 7, Braunschweig 1971,S.25-53

Pfiffner, M.: Mit Methode zu integrierten Datenbanken. Output, 9 (1980), 1, S. 19-25

Pollack, M., Wiebenson, W.: Solution of the Shortest Route Problem - A Review. Operations Research, 8 (1969), S. 224-230

Raver, N., Hubbard, G.U.: Automated Logical Data Base Design: Concepts and Applications. Informatie (NL) 20 (1978), S. 352-366

Riet van de, R.P.: Normalization - A Tool in Designing a Database. In: INFOTECH State of the Art Report. Data Design, Invited Papers. Maidenhead 1980, S. 268-281

Schlageter, G., Stucky, W.: Datenbanksysteme: Konzepte und Modelle. Stuttgart 1977

Schönenberger, P., Soldan, C.P.: Datenunabhängigkeit im Anwendungsbereich von Datenbanken. Output , 10 (1981), 6, S. 29-34

Smith, J.M., Smith, D.C.P.: Database Abstraction. In: Data Base Design Methodology, an Intensive Short Course. Engin. Summer Conf. Ann Arbor, Michigan. 1976, S. 25-35

Smith, J.M., Smith, D.C.P.: Database Abstractions: Aggregation and Generalization. ACM Transaction on Database Systems, 2 (1977), S. 105-133

Smith, J.M., Smith, D.C.P.: Conceptual Data Base Design. In: INFOTECH State of the Art Report. Data Design, Invited Papers. Maidenhead 1980, S. 139-161

Späth, H.: Cluster-Analyse-Algorithmen. München 1975

Späth, H.: Partitionierende Cluster-Analyse bei Binärdaten am Beispiel von bundesdeutschen Hochschulen und Diplomstudiengängen. Zeitschrift f. Operations Research, 21 (1977), S. B85-96

Späth, H.: Numerische Erfahrungen zu heuristischen Lösungsverfahren beim Varianzkriterium in der Cluster-Analyse. Angewandte Informatik 1977, S. 67-72

Späth, H.: Partitionierende Cluster-Analyse für große Objektmengen mit binären Merkmalen am Beispiel von Firmen und deren Berufsgruppenbedarf. In: Fallstudien zur Clusteranalyse, hrsg. von H.Späth.München 1977, S.63-80

Teorey, T.J., Fry, J.P.: Logical Database Design: A Pragmatic Approach. In: INFOTECH State of the Art Report. Data Base Technology. Maidenhead 1978, S. 357-383

Teorey, T.J., Fry, J.P.: The Logical Record Access Approach to Database Design. Computing Surveys, 12 (1980), S. 179-211

Ullman, D.J.: Principles of Database Systems. Computer Science Press. 1980

Vetter, M.: Data Base Design by Applied Data Synthesis. In: Proc. of the 3rd Int. Conf. on Very Large Data Bases. New York 1977, S. 428-440

Vetter, M.: Systematische Entwicklung von Informationssystemen. Output, 10 (1981), 6, S. 35-42

Vetter, M., Maddison, R.N.: Database Design Methodology. Engelwood Cliffs, New Jersey 1981

Wang, C., Wedekind, H.: Segment Synthesis in Logical Data Base Design. IBM Journal of Research and Development, 19 (1975), S. 71-77

Wedekind, H.: Datenbanksysteme I. Mannheim 1974

Wedekind, H.: Relationale Datenbanksysteme. Informatik-Spektrum, 1 (1978), S. 5-16

Wedekind, H., Ortner, E.: Systematisches Konstruieren von Datenbankanwendungen. München 1980

Weldon, J.-L.: Using Data Base Abstractions for Logical Design. The Computer Journal, 23 (1980), S. 41-45

Wenger, D.: Ein methodischer Weg zur konzeptionellen Datenbank. Output, 10 (1981), 1, S. 23-28

Wille, H., Gewald, K., Weber, H.: Netzplantechnik - Methoden zur Planung und Überwachung von Projekten. Band I. München 1972

Yao, S.B., Navathe, S.B., Weldon, J.L.: An Integrated Approach to Logical Database Design. In: NYU Symposium on Database Design. New York 1978

Zaniolo, C., Melkanoff, M.A.: On the Design of Relational Database Schemata. ACM Transactions on Database Systems, 6 (1981), S. 1-47

Zimmermann, G.: Typenbildung mit Methoden der Clusteranalyse. Fortschrittliche Betriebsführung, 26 (1977), S. 381-389

LEBENSLAUF

Am 28. 5.45	geboren in Blaubeuren, als Tochter des Oberstudienrats Alfred Merkle und seiner Ehefrau Gertrud, geb. Nitsch Konfession: röm.kath.
6. 9.51 - 18. 4.55	Besuch der Grundschule in Ulm
19. 4.55 - 23. 3.61	Besuch des Mädchen-Gymnasiums in Ulm
18. 4.61 - 15. 9.61	Besuch der Staatlichen Landfrauenschule (Haushaltungsschule) in Sigmaringen
1.11.61 - 31. 3.62	kaufmännisches Praktikum bei Telefunken Ag in Ulm
1. 4.62 - 31. 3.64	Lehre als Technische-Teilzeichnerin bei Telefunken AG in Ulm
1.10.64 - 14. 3.65	Werkstattpraktikum bei Klöckner-Humboldt-Deutz AG in Ulm
15. 3.65 - 12. 2.68	Studium an der Staatlichen Ingenieurschule in Ulm, Fachrichtung: Stahl-/ Leichtbau. Abschluß: Dipl.Ing. (FH)
im März 67	Schweißfachingenieur-Prüfung bei Prof. Dr. Holler in Frankfurt
15. 4.68 - 31.12.69	Statikerin im Ingenieurbüro Liebert in Ulm
1. 1.70 - 30. 6.75	Systemanalyse und Programmierung bei de Unternehmensberatung ADV ORGA, F.A. Meyer KG in Wilhelmshaven
seit 1.7.75	selbständig als Beratender Ingenieur in der Datenverarbeitung, daneben:
1. 4.75 - 23. 5.78	BWL-Studium an der Friedrich-Alexander-Universität Erlangen/Nürnberg
seit 1.8.78	Geschäftsführerin der Merkle-EDV-Beratungs GmbH Ulm

Ulm, den 23.10.82 Astrid Merkle

Hiermit versichere ich gemäß §10 Abs. 2 der Promotionsordnung der Abteilung Wirtschafts- und Sozialwissenschaften der Universität Dortmund, daß ich diese Dissertation selbständig verfaßt und mich anderer als der angegebenen Hilfsmittel nicht bedient habe.

Ulm, den 23.10.82 Astrid Merkle